

# IMPROVING THE NUMERICAL PERFORMANCE OF STATIC AND DYNAMIC AGGREGATE DISCRETE CHOICE RANDOM COEFFICIENTS DEMAND ESTIMATION

BY JEAN-PIERRE DUBÉ, JEREMY T. FOX, AND CHE-LIN SU<sup>1</sup>

The widely used estimator of [Berry, Levinsohn, and Pakes \(1995\)](#) produces estimates of consumer preferences from a discrete-choice demand model with random coefficients, market-level demand shocks, and endogenous prices. We derive numerical theory results characterizing the properties of the nested fixed point algorithm used to evaluate the objective function of BLP's estimator. We discuss problems with typical implementations, including cases that can lead to incorrect parameter estimates. As a solution, we recast estimation as a mathematical program with equilibrium constraints, which can be faster and which avoids the numerical issues associated with nested inner loops. The advantages are even more pronounced for forward-looking demand models where the Bellman equation must also be solved repeatedly. Several Monte Carlo and real-data experiments support our numerical concerns about the nested fixed point approach and the advantages of constrained optimization. For static BLP, the constrained optimization approach can be as much as ten to forty times faster for large-dimensional problems with many markets.

**KEYWORDS:** Random coefficients logit demand, constrained optimization, numerical methods, dynamics.

## 1. INTRODUCTION

DISCRETE-CHOICE DEMAND MODELS have become popular in the demand estimation literature due to their ability to accommodate rich substitution patterns between a large array of products. [Berry, Levinsohn, and Pakes \(1995; hereafter BLP\)](#) made an important contribution to this literature by accommodating controls for the endogeneity of product characteristics (namely, prices) without sacrificing the flexibility of these substitution patterns. Their methodological contribution comprises a statistical, generalized method-of-moments (GMM) estimator and a numerical algorithm for implementing this estimator.

<sup>1</sup>We thank Daniel Akerberg, Steven Berry, John Birge, Amit Gandhi, Philip Haile, Lars Hansen, Panle Jia, Kyoo il Kim, Samuel Kortum, Kenneth Judd, Sven Leyffer, Denis Nekipelov, Aviv Nevo, Jorge Nocedal, Ariel Pakes, John Rust, Hugo Salgado, Azeem Shaikh, and Richard Waltz for helpful discussions and comments. We also thank workshop participants at CREST-INSEE/ENSAE, EARIE, the ESRC Econometrics Study Group Conference, the Econometric Society, the Federal Trade Commission, INFORMS, the International Industrial Organization Conference, the 2009 NBER winter IO meetings, Northwestern University, the Portuguese Competition Commission, Santa Clara, the Stanford Institute for Theoretical Economics, the UK Competition Commission, the University of Chicago, and the University of Rochester. Dubé is grateful to the Kilts Center for Marketing and the Neubauer Faculty Fund for research support. Fox thanks the NSF, Grant 0721036, the Olin Foundation, and the Stigler Center for financial support. Su is grateful for the research support from the NSF (award no. SES-0631622) and the IBM Corporation Faculty Research Fund at the University of Chicago Booth School of Business.

As in Berry (1994), the evaluation of the GMM objective function requires inverting the nonlinear system of market share equations. BLP and Berry suggested nesting this inversion step directly into the parameter search. BLP proposed a contraction mapping to solve this system of equations. Their estimation algorithm consists of an outer-loop optimization over the structural parameters and a nested inner-loop call to the contraction mapping each time the GMM objective function is evaluated. We refer to this approach as the nested fixed point, or NFP, approach.<sup>2</sup> Following the publication of Nevo's (2000) "A Practitioner's Guide" to implementing BLP, numerous studies have emerged using BLP's algorithm for estimating discrete-choice demand systems with random coefficients.

Our first objective consists of exploring the numerical properties of BLP's nested contraction mapping algorithm. Each time the GMM objective function is evaluated, the predicted market share equation may be called hundreds or thousands of times before the contraction mapping converges. Therefore, it may be tempting to use a less stringent stopping criterion for the contraction mapping to speed up the estimation procedure. We derive theoretical results to show the adverse effects of a less stringent stopping criterion for the inner-loop on the outer-loop parameter estimates. The inner-loop error propagates into the outer-loop GMM objective function and its derivatives, which may cause an optimization routine to fail to converge. To induce the convergence of an optimization routine, the researcher may then loosen the outer-loop stopping criterion. Consequently, even when an optimization run converges, it may falsely stop at a point that is not a local minimum.

Our second objective consists of a new computational algorithm for implementing the BLP estimator. Following Su and Judd (2012), we recast the optimization of BLP's GMM objective function as a mathematical program with equilibrium constraints (MPEC). The MPEC algorithm minimizes the GMM objective function subject to the market share equations as constraints. **The smoothness of the GMM objective function and the constraints ensure that our formulation gives a standard constrained optimization problem.** From a statistical perspective, the MPEC algorithm generates the same estimator as the correctly implemented NFP approach. Therefore, the theoretical results on consistency and statistical inference in Berry, Linton, and Pakes (2004) apply to both NFP and MPEC. Our focus is on the numerical accuracy and relative speed of the two algorithms.

We prefer the MPEC algorithm over NFP for a couple of reasons. First, there is no nested inner loop and, hence, **no numerical error** from the inner-loop propagated into the objective function. This aspect eliminates the incentive for adjusting the outer-loop stopping criterion and avoids convergence to a

<sup>2</sup>We borrow the term "nested fixed point" from Rust (1987), on the estimation of dynamic programming models.

point that is not a local minimum.<sup>3</sup> Second, by eliminating the nested calls to a contraction mapping, the procedure can be faster. The potential speed advantage and capability of MPEC are enhanced further if the exact first-order and second-order derivatives are provided to the optimizer. MPEC's speed advantages can become even more pronounced for very large-dimensional problems with many markets because the constraint Jacobian and the Hessian of the Lagrangian are sparse. However, the speed advantage of MPEC will diminish for large-dimensional problems with very few markets and a large number of products because the Jacobian and Hessian are dense.

To illustrate the potential numerical issues with NFP when it is implemented poorly, we conduct an experiment with one simulated data set and one pseudo-real data set. We document cases where a loose tolerance for the contraction mapping in the NFP approach leads to incorrect parameter estimates and the failure of an optimization routine to report convergence. We observe this problem with optimization routines using closed-form and numerical derivatives. The errors in the estimated own-price elasticities are found to be large in both pseudo-real field data and simulated data. In the example with pseudo-real data, we show that the parameter estimates are always around the same incorrect point—a point that is not even a local minimum. In this case, using multiple starting points may not be able to diagnose the presence of errors in the parameter estimates. We also use this example to show that an alternative Nelder–Meade or simplex algorithm usually converges to the wrong solution.

In a second set of sampling experiments, we explore the relative performance of the correctly implemented NFP approach and MPEC. We calculate a lower bound on the Lipschitz constant of the contraction mapping that depends on the data and the parameter values from the demand model. A researcher would not be able to compute the Lipschitz constant for a given data set in practice because it depends on the model parameters. However, we can use the Lipschitz constant to guide us in the construction of Monte Carlo experiments for which MPEC becomes several times faster than NFP when we generate the data in ways that increase the Lipschitz constant. As expected, MPEC's speed is relatively invariant to the Lipschitz constant because a contraction mapping is not used. When MPEC and NFP are implemented correctly with analytic gradients and Hessians as well as their respective sparsity structures, we find that MPEC tends to be faster than NFP. As we increase the number of products, markets, and/or simulation draws, the relative speed of MPEC is enhanced to as much as ten to forty times faster than NFP. However, MPEC does not retain this speed advantage for large-dimensional, dense problems with many products and very few markets.

The theoretical results we derive can be generalized to some dynamic versions of the BLP model, where the numerical problems associated with NFP

<sup>3</sup>Petrin and Train (2010) used a control function to avoid the inner loop. Unlike MPEC, their estimator has different statistical properties than the original BLP GMM estimator.

will be magnified. Consider the recent empirical literature on durable and semidurable goods markets, where forward-looking consumers can alter the timing of their purchase decision based on expectations about future products and prices (Carranza (2010), Gowrisankaran and Rysman (2011), Hendel and Nevo (2007), Melnikov (2001), Nair (2007), Schiraldi (2011)). Estimating demand using the NFP algorithm now involves three numerical loops: the outer optimization routine, the inner inversion of the market share equations, and the inner evaluation of the consumer value function (the Bellman equations), for each heterogeneous type of consumer.

MPEC extends naturally to the case with forward-looking consumers. We optimize the statistical objective function and impose consumers' Bellman equations and market share equations as constraints. Our approach eliminates both inner loops, thereby eliminating these two sources of numerical error when evaluating the outer-loop objective function. We produce benchmark results that show that MPEC is faster than NFP under realistic data-generating processes. We expect the relative performance of MPEC to improve for more complex dynamic demand models that nest more inner loops (Lee (2010)) or for specifications that do not have a contraction mapping for inverting the share equation (Gowrisankaran and Rysman (2011)).

The remainder of the paper is organized as follows. We discuss BLP's model in Section 2 and their statistical estimator in Section 3. Section 4 provides a theoretical analysis of the NFP algorithm. Section 5 presents our alternative MPEC algorithm. Section 6 presents numerical examples of practices leading to errors in the estimates of parameters. Section 7 provides Monte Carlo evidence for the relative performances of the NFP and MPEC algorithms. Section 8 discusses the extension to a dynamic analog of BLP. We conclude in Section 9.

## 2. THE DEMAND MODEL

In this section, we present the standard random coefficients, discrete-choice model of aggregate demand. Consider a set of markets,  $t = 1, \dots, T$ , each populated by a mass  $M_t$  of consumers who each choose one of the  $j = 1, \dots, J$  products available, or opt not to purchase. Each product  $j$  is described by its characteristics  $(x_{j,t}, \xi_{j,t}, p_{j,t})$ . The vector  $x_{j,t}$  consists of  $K$  product attributes. Let  $x_t$  be the collection of the vectors  $x_{j,t}$  for all  $J$  products. The scalar  $\xi_{j,t}$  is a vertical characteristic that is observed by the consumers and firms, but is unobserved by the researcher.  $\xi_{j,t}$  can be seen as a market- and product-specific demand shock that is common across all consumers in the market. For each market, we define the  $J$ -vector  $\xi_t = (\xi_{1,t}, \dots, \xi_{J,t})$ . Finally, we denote the price of product  $j$  by  $p_{j,t}$  and the vector of the  $J$  prices by  $p_t$ .

Consumer  $i$  in market  $t$  obtains the utility from purchasing product  $j$

$$(1) \quad u_{i,j,t} = \beta_i^0 + x'_{j,t} \beta_i^x - \beta_i^p p_{j,t} + \xi_{j,t} + \varepsilon_{i,j,t}.$$

The utility of the outside good, the “no-purchase” option, is  $u_{i,0,t} = \varepsilon_{i,0,t}$ . The parameter vector  $\beta_i^x$  contains the consumer’s tastes for the  $K$  characteristics and the parameter  $\beta_i^p$  reflects the marginal utility of income,  $i$ ’s “price sensitivity.” The intercept  $\beta_i^0$  captures the value of purchasing an inside good instead of the outside good. Finally,  $\varepsilon_{i,j,t}$  is an additional idiosyncratic product-specific shock. Let  $\varepsilon_{i,t}$  be the vector of all  $J + 1$  product-specific shocks for consumer  $i$ .

Each consumer picks the product  $j$  that gives her the highest utility. If tastes,  $\beta_i = (\beta_i^0, \beta_i^x, \beta_i^p)$  and  $\varepsilon_{i,t}$ , are independent draws from the distributions  $F_\beta(\beta; \theta)$ , characterized by the parameters  $\theta$ , and  $F_\varepsilon(\varepsilon)$ , respectively, the market share of product  $j$  is

$$s_j(x_t, p_t, \xi_t; \theta) = \int_{\{\beta_i, \varepsilon_{i,t} | u_{i,j,t} \geq u_{i,j',t} \forall j' \neq j\}} dF_\beta(\beta; \theta) dF_\varepsilon(\varepsilon).$$

To simplify aggregate demand estimation, we follow the convention in the literature and assume  $\varepsilon$  is distributed Type I extreme value so that we can integrate it out analytically,

$$(2) \quad s_j(x_t, p_t, \xi_t; \theta) = \int_{\beta} \frac{\exp(\beta^0 + x'_{j,t} \beta^x - \beta^p p_{j,t} + \xi_{j,t})}{1 + \sum_{k=1}^J \exp(\beta^0 + x'_{k,t} \beta^x - \beta^p p_{k,t} + \xi_{k,t})} dF_\beta(\beta; \theta).$$

This assumption gives rise to the random coefficients logit model.

The empirical goal is to estimate the parameters  $\theta$  characterizing the distribution of consumer random coefficients,  $F_\beta(\beta; \theta)$ . For practicality, BLP assumed that  $F_\beta(\beta; \theta)$  is the product of  $K$  independent normals, with  $\theta = (\mu, \sigma)$ , the vectors of means and standard deviations for each component of the  $K$  normals.<sup>4</sup> The integrals in (2) are typically evaluated by Monte Carlo simulation with  $n_s$  draws of  $\beta$  from the distribution  $F_\beta(\beta; \theta)$ :

$$(3) \quad \hat{s}_j(x_t, p_t, \xi_t; \theta) = \frac{1}{n_s} \sum_{r=1}^{n_s} \frac{\exp(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t})}{1 + \sum_{k=1}^J \exp(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t})}.$$

In principle, other numerical integration methods could be used (Judd (1998), Chapters 7–9).

<sup>4</sup>In principle, one could proceed nonparametrically (Fox, Kim, Ryan, and Bajari (2012), Berry and Haile (2011), Chiappori and Komunjer (2009), Fox and Gandhi (2011)).

## 3. THE BLP GMM ESTIMATOR

We now briefly discuss the GMM estimator used to estimate the vector of structural parameters,  $\theta$ . To the extent that firms observe the demand shocks,  $\xi_t$ , and condition on them when they set their prices, the resulting correlation between  $p_t$  and  $\xi_t$  introduces endogeneity bias into the estimates of  $\theta$ .

BLP addressed the endogeneity of prices with a vector of instrumental variables,  $z_{j,t}$ , that are excluded from the demand equation. They proposed a GMM estimator based on the conditional moment condition  $E[\xi_{j,t} | z_{j,t}, x_{j,t}] = 0$ . The instruments  $z_{j,t}$  can be product-specific cost shifters, although frequently other instruments are used because of data availability. Usually the  $K$  non-price characteristics in  $x_{j,t}$  are also assumed to be mean independent of  $\xi_{j,t}$  and hence to be valid instruments. Computationally, the researcher often implements the moments as  $E[\xi_{j,t} \cdot h(z_{j,t}, x_{j,t})] = 0$  for some known, vector-valued function  $h$  that gives  $C$  moment conditions.

To form the empirical analog of  $E[\xi_{j,t} \cdot h(z_{j,t}, x_{j,t})]$ , the researcher needs to find the implied values of the demand shocks,  $\xi_{j,t}$ , corresponding to a guess for  $\theta$ . Let  $S_{j,t}$  be the observed market share of product  $j$  in market  $t$ . Denote  $S_t = (S_{j,t})_{j=1}^J$ . The system of market shares defines a mapping between the vector of demand shocks and the observed market shares:  $S_t = s(x_t, p_t, \xi_t; \theta)$ , or  $S_t = s(\xi_t; \theta)$  for short. Berry (1994) proved that  $s$  has an inverse,  $s^{-1}$ , such that, for a given  $\theta$ , any observed vector of shares can be explained by a unique vector  $\xi_t(\theta) = s^{-1}(S_t; \theta)$ . Let  $\xi(\theta) = (\xi_t(\theta))_{t=1}^T$  denote the vector of demand shocks in all markets. We can compute  $\xi(\theta)$  using the contraction mapping proposed in BLP. To summarize, the researcher's data consist of  $\{(x_{j,t}, p_{j,t}, S_{j,t}, z_{j,t})_{j=1}^J\}_{t=1}^T$  for  $J$  products in each of  $T$  markets.

A GMM estimator can now be constructed by using the empirical analog of the  $C$  moment conditions,

$$\begin{aligned} g(\xi(\theta)) &= \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^J \xi_{j,t}(\theta) \cdot h(z_{j,t}, x_{j,t}) \\ &= \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^J s_j^{-1}(S_t; \theta) \cdot h(z_{j,t}, x_{j,t}). \end{aligned}$$

For some weighting matrix  $W$ , we define the GMM estimator as the vector,  $\theta^*$ , that solves the problem

$$(4) \quad \min_{\theta} Q(\xi(\theta)) = \min_{\theta} g(\xi(\theta))' W g(\xi(\theta)).$$

The NFP algorithm consists of an outer-loop to minimize the GMM objective function  $Q(\xi(\theta))$  and an inner loop to evaluate this objective function at a given  $\theta$  by inverting the market share equations for  $\xi_t(\theta) = s^{-1}(S_t; \theta)$ .

The statistical efficiency of the GMM estimator can be improved by using more functions of  $z_{j,t}$  in the vector of moments, finding more instruments, using an optimal weighting matrix in a second step, or using an efficient one-step method such as continuously updated GMM or empirical likelihood.

#### 4. A THEORETICAL ANALYSIS OF THE NFP ALGORITHM

In this section, we analyze the numerical properties of the NFP algorithm. From a practical perspective, the speed of the NFP algorithm is determined by the number of calls to evaluate the objective function and the computation time associated with the inner loop for each function evaluation. While we do not know a priori how many calls to the GMM objective function will be needed, we can analyze the speed of convergence of the inner loop. We then show how numerical error from the inner loop can propagate into the outer loop, potentially leading to incorrect parameter estimates.

##### 4.1. Error in the NFP Contraction Mapping

For a given  $\theta$ , the inner loop of the NFP algorithm solves the share equations  $S_t = s(\xi_t; \theta)$  for the demand shocks  $\xi(\theta)$  by iterating the contraction mapping

$$(5) \quad \xi_t^{h+1} = \xi_t^h + \log S_t - \log s(\xi_t^h; \theta), \quad t = 1, \dots, T,$$

until the successive iterates  $\xi_t^{h+1}$  and  $\xi_t^h$  are sufficiently close.<sup>5</sup> Formally, one chooses a small number, for example,  $10^{-8}$  or  $10^{-14}$ , for  $\epsilon_{\text{in}}$  as the inner-loop tolerance level and requires  $\xi_t^{h+1}$  and  $\xi_t^h$  to satisfy the stopping rule

$$(6) \quad \|\xi_t^h - \xi_t^{h+1}\| \leq \epsilon_{\text{in}}$$

for the iteration  $h + 1$  when we terminate the contracting mapping (5). Let  $\xi_t(\theta, \epsilon_{\text{in}})$  denote the first  $\xi_t^{h+1}$  such that the stopping rule (6) is satisfied. We then use  $\xi_t(\theta, \epsilon_{\text{in}})$  to approximate  $\xi_t(\theta)$ .

The contraction mapping theorem (e.g., Dahlquist and Björck (2008)) provides a bound on the error in (5)

$$\|\xi^h - \xi(\theta)\| \leq \|\xi^h - \xi^{h-1}\| \frac{L(\theta)}{1 - L(\theta)} \leq \|\xi^1 - \xi^0\| \frac{L(\theta)^h}{1 - L(\theta)}.$$

The Lipschitz constant,  $L(\theta)$ , characterizes an upper bound involving the number of calls to the share equation.

<sup>5</sup>In our implementation of NFP, to conform with Nevo (2000) we iterate over  $\exp(\xi)$ . However, depending on the magnitude of  $\xi$ , the use of the exponentiated form  $\exp(\xi)$  in a contraction mapping can lose three to five digits of accuracy in  $\xi$ , and as a result, introduce an additional source of numerical error. For example, if  $|\xi_t^h| = -8$  and  $|\exp(\xi_t^h) - \exp(\xi_t^{h+1})| = 10^{-10}$ , then  $|\xi_t^h - \xi_t^{h+1}| = 2.98 \times 10^{-7}$ .



For the inner loop, BLP defined the Lipschitz constant as  $L(\theta) = \max_{\xi \in \Xi} \|I - \nabla_{\xi}(\log s(\xi; \theta))\|_{\infty}$ ,<sup>6</sup> where

$$\frac{\partial \log(s_j(\xi_t; \theta))}{\partial \xi_{lt}} = \begin{cases} \frac{\sum_{r=1}^{n_s} [\mathcal{T}_j(\xi_t; \theta^r) - \mathcal{T}_j(\xi_t; \theta^r)^2]}{\sum_{r=1}^{n_s} \mathcal{T}_j(\xi_t; \theta^r)}, & \text{if } j = l, \\ -\frac{\sum_{r=1}^{n_s} [\mathcal{T}_j(\xi_t; \theta^r) \mathcal{T}_l(\xi_t; \theta^r)]}{\sum_{r=1}^{n_s} \mathcal{T}_j(\xi_t; \theta^r)}, & \text{if } j \neq l, \end{cases}$$

and  $\mathcal{T}_j(\xi_t; \theta^r) = \frac{\exp(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t})}{1 + \sum_{k=1}^J \exp(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t})}$ . BLP proved that  $L(\theta) < 1$ . The Lipschitz constant is related to the matrix of own and cross demand elasticities with respect to the demand shocks,  $\xi$ . In Section 7, we use the Lipschitz constant to distinguish between simulated datasets where we expect the contraction mapping to perform relatively slowly and rapidly.

#### 4.2. Ensuring Convergence for the Outer Loop in NFP

In this subsection, we show how inner-loop numerical error propagates into the GMM objective function and its gradient, and characterize the corresponding numerical inaccuracy.

For a given  $\theta$ , we denote by  $\xi(\theta, 0)$  and  $\xi(\theta, \epsilon_{\text{in}})$  the true demand shocks and the approximated demand shocks with an inner-loop tolerance  $\epsilon_{\text{in}}$ , respectively. Let  $Q(\xi(\theta, \epsilon_{\text{in}}))$  be the GMM objective function with the inner-loop tolerance  $\epsilon_{\text{in}}$ . For  $\epsilon_{\text{in}} = 0$ , we use the compact notation  $Q(\xi(\theta)) = Q(\xi(\theta, 0))$ .

The following lemma states that the biases in evaluating the GMM objective function and its gradient at a vector of structural parameters are of the same order as the inner-loop tolerance, adjusted by the Lipschitz constant for the inner-loop contraction mapping.

**LEMMA 1:** *Let  $L(\theta)$  be the Lipschitz constant for the inner-loop contraction mapping at structural parameters  $\theta$ . Given an inner-loop tolerance  $\epsilon_{\text{in}}$ ,*

- (i)  $|Q(\xi(\theta, \epsilon_{\text{in}})) - Q(\xi(\theta, 0))| = O(\frac{L(\theta)}{1-L(\theta)} \epsilon_{\text{in}})$ ,
- (ii)  $\|\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\theta, \epsilon_{\text{in}})} - \nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\theta, 0)}\| = O(\frac{L(\theta)}{1-L(\theta)} \epsilon_{\text{in}})$ ,

*assuming both  $\|\frac{\partial Q(\xi(\theta))}{\partial \xi}|_{\xi=\xi(\theta, 0)}\|$  and  $\|\frac{\partial \nabla_{\theta} Q(\xi(\theta))}{\partial \xi}|_{\xi=\xi(\theta, 0)}\|$  are bounded.*

<sup>6</sup>The matrix norm  $\|\cdot\|_{\infty}$  is used in BLP (1995).



The proof is in Appendix A.

Given the inner-loop tolerance,  $\epsilon_{\text{in}}$ , and the corresponding inner-loop error, an optimization solver is searching for a vector of structural parameters  $\hat{\theta}(\epsilon_{\text{in}})$  such that  $\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})} = 0$ . The notation  $\hat{\theta}(\epsilon_{\text{in}})$  indicates its dependence on the inner-loop tolerance  $\epsilon_{\text{in}}$ . In practice, convergence is achieved when the norm of the gradient,  $\|\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})}\|$ , at some parameters  $\tilde{\theta}$  is smaller than a predetermined outer-loop tolerance,  $\epsilon_{\text{out}}$ :  $\|\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})}\| \leq \epsilon_{\text{out}}$ . The next theorem gives a bound on the gradient of the GMM objective function at  $\tilde{\theta}$  in the neighborhood of  $\hat{\theta}(\epsilon_{\text{in}})$ .

**THEOREM 2:** *Given the inner-loop tolerance  $\epsilon_{\text{in}}$ , let  $\hat{\theta}(\epsilon_{\text{in}})$  be a vector of structural parameters such that  $\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})} = 0$ . For any  $\tilde{\theta}$  near  $\hat{\theta}(\epsilon_{\text{in}})$ ,*

$$\begin{aligned} & \|\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})}\| \\ & \leq O\left(\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1 - L(\hat{\theta}(\epsilon_{\text{in}}))} + \frac{L(\tilde{\theta})}{1 - L(\tilde{\theta})}\right)\epsilon_{\text{in}} + \|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\|\right), \end{aligned}$$

assuming  $\|\nabla_{\theta}^2 Q(\xi(\theta))|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)}\|$  is bounded.

The proof is in Appendix A. Theorem 2 indicates that, for the outer-loop GMM minimization to report convergence around  $\hat{\theta}(\epsilon_{\text{in}})$ , the outer-loop tolerance  $\epsilon_{\text{out}}$  should be chosen to be of the same order as  $\epsilon_{\text{in}}$ .

Researchers often use numerical gradients, which introduce an additional source of bias. We can use Lemma 9.1 in Nocedal and Wright (2006) to characterize the bound on the error of the gradient when the optimizer uses a central difference derivative  $\nabla_d Q(\xi(\theta, \epsilon_{\text{in}}))$  with difference interval  $d$ :

$$\begin{aligned} (7) \quad & \|\nabla_d Q(\xi(\theta, \epsilon_{\text{in}})) - \nabla_{\theta} Q(\xi(\theta))\|_{\infty} \\ & \leq O(d^2) + \frac{1}{d} \sup_{\tilde{\theta}: \|\tilde{\theta} - \theta\| \leq d} |Q(\xi(\tilde{\theta}, \epsilon_{\text{in}})) - Q(\xi(\tilde{\theta}, 0))|. \end{aligned}$$

The  $O(d^2)$  term represents the error from a finite difference approximation. The second “noise” term arises from the numerical error in the objective function for a given  $\epsilon_{\text{in}}$ . It is ambiguous whether reducing the difference interval,  $d$ , reduces the numerical error in the gradient. However, Nocedal and Wright noted that “if the noise term dominates the difference interval ... it will only be pure luck if  $[-\nabla_d Q(\xi(\theta, \epsilon_{\text{in}}))]$  turns out to be a direction of descent for  $[Q]$ .” It also follows from Theorem 2 that if the bound on the error in the gradient is large, one may need to choose a larger outer-loop tolerance,  $\epsilon_{\text{out}}$ , to induce the solver to report convergence. We illustrate these points in our first empirical example in Section 6. We show that the combination of a loose inner loop, numerical derivatives, and a loose outer-loop tolerance (to ensure

convergence) produces estimates that are not a local optimum of the objective function. Given that closed-form derivatives can be constructed for static BLP, we recommend that users avoid numerical derivatives.

In practice, we have found cases where the BLP algorithm was implemented with loose inner-loop tolerances,  $\epsilon_{\text{in}}$ , to speed up the convergence of the contraction mapping.<sup>7</sup> The resulting imprecision in the gradient could prevent the optimization routine from detecting a local minimum and converging. In turn, the researcher may need to use a loose outer-loop tolerance, such as  $\epsilon_{\text{out}} = 10^{-2}$ , to ensure the convergence of the outer-loop optimization. Raising  $\epsilon_{\text{out}}$  reduces precision in the estimates and, worse, could generate an estimate that is not a local minimum. In our Monte Carlo experiments below, we illustrate how a loose stopping criterion for the outer loop can cause the optimization routine to terminate early or produce incorrect point estimates that do not satisfy the first-order conditions for a local minimizer.

The default value for  $\epsilon_{\text{out}}$  in most optimization solvers is  $10^{-6}$ . Since we do not know the exact magnitude on the bound of the gradient, we choose  $\epsilon_{\text{in}} = 10^{-14}$  in our implementation. This is a conservative choice of tolerance that will ensure that convergence in the outer-loop optimization problem is reliable and robust.

#### 4.3. Bias in Parameter Estimates From the Inner-Loop Numerical Error

In this section, we discuss the small-sample biases associated with inner-loop numerical error. Let  $\theta^* = \arg \min_{\theta} \{Q(\xi(\theta), 0)\}$  be the minimizer of the finite-sample objective function without the inner-loop error. We study the bound on the bias in the estimates,  $\hat{\theta}(\epsilon_{\text{in}}) - \theta^*$ , from using an inner-loop tolerance  $\epsilon_{\text{in}}$ .

**THEOREM 3:** *Assuming  $\left\| \frac{\partial Q(\xi(\theta))}{\partial \xi} \right\|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \parallel$  is bounded, the following inequality holds:*

$$O(\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\|^2) \leq |Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\theta^*, 0))| + O\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1 - L(\hat{\theta}(\epsilon_{\text{in}}))} \epsilon_{\text{in}}\right).$$

<sup>7</sup>Some researchers have customized an adaptive version of the inner-loop tolerance. The procedure consists of using a loose inner-loop tolerance when the parameter estimates appear “far” from the solution, and switching to a tighter inner-loop tolerance when the parameter estimates are “close” to the solution. The switch between the loose and tight inner-loop tolerances is usually based on the difference between the successive parameter iterates: for example, if  $\|\theta^{k+1} - \theta^k\| \leq 0.01$ , then  $\epsilon_{\text{in}} = 10^{-8}$ ; otherwise,  $\epsilon_{\text{in}} = 10^{-6}$ . Suppose that the following sequence of iterates occurs:  $\|\theta^{k+1} - \theta^k\| \geq 0.01$  ( $\epsilon_{\text{in}} = 10^{-6}$ ),  $\|\theta^{k+2} - \theta^{k+1}\| \leq 0.01$  ( $\epsilon_{\text{in}} = 10^{-8}$ ), and  $\|\theta^{k+2} - \theta^{k+1}\| \geq 0.01$  ( $\epsilon_{\text{in}} = 10^{-6}$ ). The NFP objective value can oscillate because of the use of two different inner-loop tolerances. This oscillation can prevent the NFP approach from converging.

The proof is in Appendix A. The first term on the right-hand side of the inequality is the bias in the GMM function evaluated at the finite-sample true and estimated parameter values. The second term arises from  $\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)$ , the bias in the demand shocks with and without inner-loop error.

The bound in Theorem 3 is not always sharp. Akerberg, Geweke, and Hahn (2009, Theorem 2) studied the case where the approximated objective function is differentiable in parameters that are similar to the inner-loop tolerance,  $\epsilon_{\text{in}}$ , and found a tighter bound. However, in the NFP context, the approximated GMM objective function,  $Q(\xi(\theta, \epsilon_{\text{in}}))$ , is not differentiable with respect to the inner-loop tolerance  $\epsilon_{\text{in}}$  because a continuous change in  $\epsilon_{\text{in}}$  can cause a discrete change in the approximate demand shocks  $\xi(\theta, \epsilon_{\text{in}})$  and hence the GMM objective function  $Q(\xi(\theta, \epsilon_{\text{in}}))$ . Furthermore, we conjecture that, for a fixed  $\epsilon_{\text{in}} > 0$ ,  $\xi(\theta, \epsilon_{\text{in}})$  and  $Q(\xi(\theta, \epsilon_{\text{in}}))$  are not differentiable with respect to  $\theta$ . In our proof for Theorem 3, we take this nondifferentiability into account and obtain a square-root upper bound.

The numerical error from the inner loop persists even in large samples. Increasing the sample size  $T \rightarrow \infty$  will not cause  $\hat{\theta}(\epsilon_{\text{in}}) \rightarrow \theta_0^*$ , where  $\theta_0^*$  is the true parameter in the data-generating process.

## 5. MPEC: A CONSTRAINED OPTIMIZATION APPROACH

In this section, we propose an alternative algorithm to compute the GMM estimator based on Su and Judd's (2012) constrained optimization approach, MPEC. Su and Judd used this approach to estimate single-agent dynamic discrete models. We use this same insight to solve for the fixed point associated with the inversion of market shares.

Let  $W$  be the GMM weighting matrix. Our constrained optimization formulation is

$$(8) \quad \min_{\theta, \xi} g(\xi)' W g(\xi) \\ \text{subject to } s(\xi; \theta) = S.$$

The moment condition term is  $g(\xi) = \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^J \xi_{j,t} \cdot h(z_{j,t}, x_{j,t})$ . The market share equations are imposed as nonlinear constraints to the optimization problem. The objective function is a quadratic function of the demand shocks  $\xi$ . We optimize over both the demand shocks  $\xi$  and the structural parameters  $\theta$ .

Su and Judd (2012) showed that the MPEC and NFP algorithms compute the same statistical estimator. Hence, any statistical property of the original BLP estimator applies to the estimator when computed via MPEC. The following proposition further indicates that the first-order conditions of the GMM problem (4) and the MPEC (8) are equivalent.

**PROPOSITION 4:** *Let  $\xi^* = \xi(\theta^*)$  be the corresponding demand shock at given structural parameters  $\theta^*$ . The vector  $(\theta^*, \xi^*)$  satisfies the first-order conditions to*

the MPEC (8) if and only if  $\theta^*$  satisfies the first-order conditions of the GMM problem (4).

The proof appears in Appendix A.

The MPEC defined by (8) can be solved using a modern nonlinear optimization package. There are two main advantages of MPEC relative to NFP. First, there is no inner-loop error that propagates into the objective function. For MPEC, the error in the estimates is of the same order as the optimality and feasibility tolerances (see Robinson (1974), Theorem 2.2).<sup>8</sup> Researchers do not need to adjust the default settings on feasibility and optimality tolerances to induce the solver to report convergence. In contrast, NFP requires a customized nested fixed point calculation, including the choice of tolerance, which could result in potential errors. As we showed previously, loosening the inner-loop to speed computation could require loosening the outer-loop to ensure that convergence is reported by the solver. Second, MPEC may be faster than NFP. One source of speed improvement comes from the fact that MPEC allows constraints to be violated during optimization. Modern optimization solvers do not enforce that the constraints are satisfied at every iteration. The constraints only need to hold at a solution. This feature allows MPEC to avoid wasting computational time on iterates far away from the true parameters. In contrast, the NFP algorithm requires solving the share equation (3) *exactly* for every  $\theta$  examined in the optimization outer-loop. Consequently, the share equation needs to be evaluated hundreds or thousands of times at each  $\theta$  examined.

One potential criticism of the MPEC approach is that it embodies a much larger-dimensional optimization problem, which could offset the speed benefit mentioned above. This concern can be addressed by exploiting the sparsity structure of the Jacobian of the market share equations: the demand shocks for market  $t$  do not enter the constraints for other markets  $t' \neq t$ . We can exploit sparsity even further by treating the moments as additional parameters and reformulating the problem in (8) as

$$(9) \quad \min_{\theta, \xi, \eta} \eta' W \eta$$

$$\text{subject to } g(\xi) = \eta,$$

$$s(\xi; \theta) = S.$$

It is easy to see that the two formulations, (8) and (9), are equivalent. The additional constraint  $g(\xi) - \eta = 0$  is linear and, hence, does not increase computational difficulty. The advantage with this alternative formulation is that, by introducing additional variables and linear constraints, the Hessian of the Lagrangian function is sparse. In general, supplying exact first-order and second-order derivatives to the optimizer will decrease computational time substantially. In addition, the capability of MPEC will be enhanced further when the

<sup>8</sup>We thank Jorge Nocedal for pointing to us this result and the reference.

sparsity patterns of first-order (Jacobian) and second-order (Hessian) derivatives are provided to the optimizer. Exploiting the sparsity of the optimization problem will both increase the speed of MPEC and enable MPEC to handle larger-dimensional problems. We discuss the speed improvements associated with sparsity and analytic Hessians and constraint Jacobians in the Supplemental Material (Dubé, Fox, and Su (2012)), where we also provide formal derivations. In our Monte Carlo experiments below, we show that increasing the sample size and, hence, the dimension of the optimization problem, does not appear to disadvantage MPEC relative to NFP, except for very large-dimensional, dense problems with very few markets and a very large number of product alternatives. The applicability of MPEC is not limited to GMM estimation. In the Supplemental Material, we present an MPEC formulation for likelihood-based approaches that require inverting the market share system and, hence, have typically been estimated using NFP.

## 6. PARAMETER ERRORS FROM LOOSE INNER-LOOP TOLERANCES IN NFP

In this section, we provide empirical support for the numerical issues discussed in Section 4. We present examples in which parameter errors can arise both in the context of sampling experiments and with pseudo-real field data.

### 6.1. Synthetic Data, Numerical Derivatives, and False Parameter Estimates

In this section, we use a single synthetic dataset to study the performance of different implementations of NFP. Details about the data-generating process are provided in Appendix B. For estimation, we use a one-step GMM estimator with the weighting matrix  $W = (Z'Z)^{-1}$ , where  $Z$  is the matrix of instruments. Our NFP code is written in the MATLAB programming environment. We use the TOMLAB interface to call the KNITRO optimization package (Byrd, Hribar, and Nocedal (1999), Byrd, Nocedal, and Waltz (2006)) in MATLAB.<sup>9</sup>

We compare three implementations of the NFP algorithm, each initialized with the same starting values. In the first scenario, “loose inner,” we use a tight outer-loop tolerance,  $\epsilon_{\text{out}} = 10^{-6}$ , and a loose inner-loop tolerance,  $\epsilon_{\text{in}} = 10^{-4}$ . In the second scenario, “loose both,” we use a loose outer-loop tolerance,  $\epsilon_{\text{out}} = 10^{-2}$ , and a loose inner-loop tolerance,  $\epsilon_{\text{in}} = 10^{-4}$ . We think of this scenario as representing the attempt of the researcher to loosen the outer-loop to promote convergence. In practice, the converged point may not actually satisfy the first-order conditions. In the third scenario, “tight both,” we implement best practice settings for the NFP algorithm with  $\epsilon_{\text{in}} = 10^{-14}$  and  $\epsilon_{\text{out}} = 10^{-6}$ . We construct 100 independent starting values for the nonlinear model parameters  $\text{SD}[\beta_i]$  by drawing them from a uniform distribution  $U(0, 7)$ . We run each

<sup>9</sup>We found that MATLAB's included solvers, `fminunc` and `fmincon`, often fail to converge to a local minimum.

TABLE I  
THREE NFP IMPLEMENTATIONS: VARYING STARTING VALUES FOR ONE SYNTHETIC DATA SET,  
WITH NUMERICAL DERIVATIVES<sup>a</sup>

|                                | NFP<br>Loose Inner | NFP<br>Loose Both | NFP<br>Tight | Truth |
|--------------------------------|--------------------|-------------------|--------------|-------|
| Fraction convergence           | 0.0                | 0.54              | 0.95         |       |
| Frac. < 1% > “global” min.     | 0.0                | 0.0               | 1.00         |       |
| Mean own-price elasticity      | −7.24              | −7.49             | −5.77        | −5.68 |
| Std. dev. own-price elasticity | 5.48               | 5.55              | ~0           |       |
| Lowest objective               | 0.0176             | 0.0198            | 0.0169       |       |

<sup>a</sup>We use 100 starting values for one synthetic data set. The NFP loose inner-loop implementation has  $\epsilon_{\text{in}} = 10^{-4}$  and  $\epsilon_{\text{out}} = 10^{-6}$ . The NFP loose-both implementation has  $\epsilon_{\text{in}} = 10^{-4}$  and  $\epsilon_{\text{out}} = 10^{-2}$ . The NFP-tight implementation has  $\epsilon_{\text{in}} = 10^{-14}$  and  $\epsilon_{\text{out}} = 10^{-6}$ . We use numerical derivatives using KNITRO’s built-in procedures.

of the three NFP implementations with these starting values, using numerical derivatives in the outer-loop optimization.

We report the results in Table I. The first row reports the fraction of runs for which the routine reports convergence. Supporting Theorem 2, we find in column one that the optimization routine will never report convergence if the inner-loop tolerance is loose,  $\epsilon_{\text{in}} = 10^{-4}$ , even when the outer-loop tolerance has the default tight tolerance of  $\epsilon_{\text{out}} = 10^{-6}$ . In contrast, column two indicates that the algorithm is more likely to converge (54% of the runs) when we also loosen the tolerance on the outer-loop. Below, we will show that convergence in this case is misleading; the estimates are far from the truth. Finally, NFP with tight tolerances converges in 95% of the runs.

To diagnose the quality of the estimates, the second row of Table I shows the fraction of runs where the reported GMM objective function value was within 1% of the lowest objective function that we found across all three NFP implementations and all 100 starting values for each case. In the first two columns, corresponding to the scenario with a loose inner loop and the scenario with a loose inner loop and a loose outer loop, respectively, none of the 100 starting values produced the lowest objective value. In contrast, NFP tight found the lowest objective value in each of the 100 runs.

The third and fourth rows of Table I report the estimated mean and standard deviation of the own-price elasticities across products, observations, and starting values to show how naive implementations could produce misleading economic predictions. The final column of row three reports the own-price elasticity of demand evaluated at the true parameter values: −5.68. As expected, NFP with a tight tolerance produces an estimate near the truth, −5.77, but our two loose implementations do not. The mean of the NFP loose-inner implementation is −7.24, higher in absolute value than the true value of −5.68. The loose-both mean is −7.49. The standard deviations of own-price elasticities for the loose inner-loop tolerances are huge: 5.48 and 5.55. The loose imple-

mentations often fail to converge to a local optimum and produce unreliable estimates that are very sensitive to starting values. We do not observe these problems with tight tolerances.

### 6.2. Parameter Errors With Nevo's Data and Closed-Form Derivatives

In this section, we apply the three implementations of NFP from the last section to the pseudo-real cereal dataset from [Nevo \(2000\)](#). For this example, we use closed-form derivatives, which should improve the performance of all three NFP implementations. For each of the three implementations (loose inner, loose both, and tight both), we use the same set of 50 starting values.<sup>10</sup>

We report the results in Table II. In row one, we find that none of the loose-inner starting values converge. In the second column, we find that loosening the outer-loop increases the convergence rate to 76% of the starting values. All of the starting values converge for NFP tight. The second row shows that all of the NFP-tight starting values also find the same local minimum with objective value 0.00202. In contrast, neither the loose-inner nor loose-both implementations of NFP find this solution, terminating at points with higher objective value (especially the loose-both implementation). We inspected results from all 50 runs and found that only one of the 50 runs using the loose inner loop was anywhere near the best solution for NFP tight.

The loose-inner and loose-both implementations find a mean own-price elasticity of  $-3.82$  and  $-3.69$ , respectively. These are about half the value of

TABLE II  
THREE NFP IMPLEMENTATIONS: VARYING STARTING VALUES FOR NEVO'S CEREAL DATA SET,  
WITH CLOSED-FORM DERIVATIVES<sup>a</sup>

|   | NFP<br>Loose Inner | NFP<br>Loose Both | NFP<br>Tight Both | NFP Tight<br>Simplex |
|---|--------------------|-------------------|-------------------|----------------------|
| Fraction reported convergence                   | 0.0                | 0.76              | 1.00              | 1.00                 |
| Frac. obj. fun. < 1% greater than "global" min. | 0.0                | 0.0               | 1.00              | 0.0                  |
| Mean own-price elasticity across all runs       | -3.82              | -3.69             | -7.43             | -3.84                |
| Std. dev. own price elasticity across all runs  | 0.4                | 0.07              | ~0                | 0.35                 |
| Lowest objective function value                 | 0.00213            | 0.00683           | 0.00202           | 0.00683              |
| Elasticity for run with lowest obj. value       | -6.71              | -3.78             | -7.43             | -3.76                |

<sup>a</sup>We use the same 50 starting values for each implementation. The NFP loose inner-loop implementation has  $\epsilon_{\text{in}} = 10^{-4}$  and  $\epsilon_{\text{out}} = 10^{-6}$ . The NFP loose-both implementation has  $\epsilon_{\text{in}} = 10^{-4}$  and  $\epsilon_{\text{out}} = 10^{-2}$ . The NFP-tight implementation has  $\epsilon_{\text{in}} = 10^{-14}$  and  $\epsilon_{\text{out}} = 10^{-6}$ . The Nelder-Meade or simplex method uses a tighter inner-loop tolerance of  $\epsilon_{\text{in}} = 10^{-14}$  and MATLAB's default values for the simplex convergence criteria. We manually code closed-form derivatives for all methods other than for Nelder-Meade, which does not use derivative information.

<sup>10</sup>We pick our starting values by taking 50 draws from the standard normal distribution. We also experimented with multiplying the starting values by the solution reported in [Nevo \(2000\)](#). The results were similar.



−7.43 found with NFP tight. Because all 50 starting values of NFP with tight settings converge to the same point, we see no variance in its elasticity estimates. In contrast, the loose inner-loop implementations are very sensitive to starting values, leading to standard deviations of 0.40 and 0.07.

Recent work has critiqued the robustness of BLP's GMM estimator. Using these same data, Knittel and Metaxoglou (2008) found that many solvers stopped at different points and that these points were often sensitive to start values. Nevo's (2000) "Researcher's Guide" reported estimates that were not a local optimum because of the use of a loose outer-loop tolerance. Using our implementation above with a state-of-the-art solver and 50 starting points, we found the same local minimum each run. Moreover, our local minimum coincided with the point with the lowest objective function value, 0.00202, reported by Knittel and Metaxoglou. We conclude that with multiple starting values, careful implementation of the numerical procedures (a tight inner loop and tight outer loop), and a state-of-the-art optimization solver, the BLP estimator produces reliable estimates.

Other work has used non-derivative-based solvers. We use the Nelder–Meade, or simplex algorithm, using the same 50 starting values as above and a tight inner loop and a tight outer loop. In column four of Table II, none of the 50 runs of the simplex algorithm finds the global minimum. Moreover, none of these runs satisfies the first-order optimality conditions. The point estimates using Nelder–Meade are systematically different (and at a higher objective function value) from the NFP tight implementation using a gradient-based search. The elasticity estimate of −3.76 is around half of the −7.43 found with tight NFP settings, and the elasticity's standard deviation across the 50 starting values is a relatively tight 0.35. See McKinnon (1998) and Wright (1996) for discussion of the problems with the Nelder–Meade algorithm.

## 7. SPEED COMPARISONS OF MPEC AND NFP

In this section, we run several Monte Carlo experiments using synthetic data to compare the speed of MPEC and NFP with a tight inner-loop tolerance. Details about the data-generating process are provided in Appendix B. We construct several sampling experiments that vary the mean of the intercept,  $E[\beta_i^0]$ , from −2 to 4, which also varies the Lipschitz constant from 0.90 to 0.99. The shares of the outside good range from 91% to 35%, consistent with most empirical applications of BLP.<sup>11</sup>

We use the MPEC formulation in (9). Similarly to our NFP implementation, the MPEC estimation code is written in the MATLAB programming environment and uses the TOMLAB interface to call KNITRO. We derive and code the closed-form first-order and second-order derivatives for both NFP

<sup>11</sup>In the original autos application in BLP (1995), the share of the outside good is around 90% in every year of the data. In Nevo (2001), the mean outside share across markets is 52%.

and MPEC.<sup>12</sup> We also supply the sparsity patterns of the constraint Jacobian and of the Hessian of the Lagrangian to the optimization routine for MPEC.<sup>13</sup> The derivation of first-order and second-order derivatives for NFP and MPEC and the relevant sparsity patterns are provided in the Supplemental Material. For both NFP and MPEC, we use the default optimality tolerance  $10^{-6}$  in KNITRO. For MPEC, we use the default feasibility tolerance  $10^{-6}$ . As mentioned above, the inner-loop tolerance for NFP is  $\epsilon_{\text{in}} = 10^{-14}$ . For all implementations of NFP and MPEC, we use the interior point algorithm with algorithm option  $\text{ALG} = 1$  in KNITRO, which is a direct decomposition of the first-order Karush Kuhn Tucker (KKT) matrix. For detailed descriptions of algorithm options in KNITRO, see [Waltz and Plantenga \(2009\)](#). One could probably further increase the speed performance of both NFP and MPEC with parallelization, but this is beyond the scope of the current analysis.

We report our results in Table III. In all cases, the lowest objective function corresponded to a case where the algorithm reported that a locally optimal solution had been found. We assess the estimates by looking at the mean own-price elasticities. For each algorithm, we also report the total CPU time required across all the five starting values. All numbers in Table III are means across 20 replications, each with a new dataset.

As expected, NFP and MPEC converged in all scenarios. MPEC and NFP almost always generated identical point estimates. We compute the root mean-squared error (RMSE) and the bias of the own-price elasticities. In all cases, the RMSE is low and the bias is moderate at around 0.08, in comparison with a base elasticity of around  $-10.5$ , suggesting that the BLP estimator is capable of recovering true demand elasticities.

Run times for NFP tight vary with the level of the Lipschitz constant. For the lower Lipschitz case with  $E[\beta_i^0] = -2$ , the average run time across the 20 replications is roughly 20 minutes for NFP and for MPEC. As we increase  $E[\beta_i^0]$ , the run times for NFP increase, while the run times for MPEC change little. When  $E[\beta_i^0] = 4$ , the highest Lipschitz case, a single run with five starting values of NFP takes, on average, 5 hours, whereas MPEC takes only 12 minutes. MPEC is 24 times faster. Table IV provides the underlying intuition for the source of the speed improvement. For both NFP and MPEC, we see almost no change across the various experiments in the number of major iterations, function evaluations, and gradient and Hessian evaluations. Furthermore, NFP requires fewer iterations and function evaluations than MPEC.<sup>14</sup> As indicated in the last column in Table IV, the main reason for NFP's time increases is

<sup>12</sup>In the Supplemental Material, we show that supplying these derivatives increases the speed of both algorithms 3 to 10 times.

<sup>13</sup>In our examples, using sparse matrices in the analytic derivatives and providing the sparsity patterns reduce memory use by 90%.

<sup>14</sup>The function evaluations for both algorithms consist of the objective function, gradients, and Hessian. The function evaluations for MPEC also include the constraints.

TABLE III  
MONTE CARLO RESULTS VARYING THE LIPSCHITZ CONSTANT<sup>a</sup>

| Intercept<br>$E[\beta_t^0]$ | Lipsch.<br>Const. | Imple.    | Runs<br>Converged | CPU Time<br>(min) | Elasticities |      |       | Outside<br>Share |
|-----------------------------|-------------------|-----------|-------------------|-------------------|--------------|------|-------|------------------|
|                             |                   |           |                   |                   | Bias         | RMSE | Truth |                  |
| −2                          | 0.891             | NFP-tight | 100%              | 21.7              | −0.077       | 0.14 | −10.4 | 0.91             |
|                             |                   | MPEC      | 100%              | 18.3              | −0.076       | 0.14 |       |                  |
| −1                          | 0.928             | NFP-tight | 100%              | 28.3              | −0.078       | 0.15 | −10.5 | 0.86             |
|                             |                   | MPEC      | 100%              | 16.3              | −0.077       | 0.15 |       |                  |
| 0                           | 0.955             | NFP-tight | 100%              | 41.7              | −0.079       | 0.16 | −10.6 | 0.79             |
|                             |                   | MPEC      | 100%              | 15.2              | −0.079       | 0.16 |       |                  |
| 1                           | 0.974             | NFP-tight | 100%              | 71.7              | −0.083       | 0.16 | −10.7 | 0.69             |
|                             |                   | MPEC      | 100%              | 11.8              | −0.083       | 0.16 |       |                  |
| 2                           | 0.986             | NFP-tight | 100%              | 103               | −0.085       | 0.17 | −10.8 | 0.58             |
|                             |                   | MPEC      | 100%              | 13.5              | −0.085       | 0.17 |       |                  |
| 3                           | 0.993             | NFP-tight | 100%              | 167               | −0.088       | 0.17 | −11.0 | 0.46             |
|                             |                   | MPEC      | 100%              | 10.7              | −0.088       | 0.17 |       |                  |
| 4                           | 0.997             | NFP-tight | 100%              | 300               | −0.091       | 0.16 | −11.0 | 0.35             |
|                             |                   | MPEC      | 100%              | 12.7              | −0.090       | 0.16 |       |                  |

<sup>a</sup>There are 20 replications for each experiment and reported means are across these 20 replications. Each replication uses five starting values to do a better job at finding a global minimum. For each NFP starting value, we run the inner-loop once and use this vector of demand shocks and mean taste parameters as starting values for MPEC. The NFP-tight implementation has  $\epsilon_{\text{in}} = 10^{-14}$  and  $\epsilon_{\text{out}} = 10^{-6}$ . There is no inner-loop in MPEC;  $\epsilon_{\text{out}} = 10^{-6}$  and  $\epsilon_{\text{feasible}} = 10^{-6}$ . The same 1000 simulation draws are used to generate the data and to estimate the model. NFP and MPEC use closed-form first- and second-order derivatives. We supply the sparsity patterns of the constraints and derivatives to the optimization routine for both methods.

that the total number of contraction mapping iterations and, hence, the number of evaluations of market share equations, increases dramatically with the Lipschitz constant.

We conduct several additional Monte Carlo experiments to ensure that the relative performance of MPEC is robust to larger datasets and to more simulation draws. Table V reports the results from varying the number of markets  $T$ , products  $J$ , and simulation draws  $n_s$ . MPEC’s relative speed advantage is not only robust to these large problems, it appears to be enhanced; MPEC is about ten to forty times faster than NFP. For 250 markets, 25 products, and 3000 simulation draws, NFP took 80 hours or 3.3 days to complete five runs, while MPEC took only 3 hours. The results in Table VI indicate that NFP requires fewer major iterations and function evaluations than MPEC. However, NFP requires hundreds of thousands of calls to the predicted market share equations. The sparsity structure is critical for these large-dimensional problems. In the Supplemental Material, we show that MPEC can handle a problem with  $T = 2000$  markets and  $J = 15$  products using only 500 MB of RAM when sparsity information is supplied. The program runs out of memory on a computer with 12 GB of RAM when sparsity information is not supplied.

TABLE IV  
ADDITIONAL RESULTS FOR VARYING THE LIPSCHITZ CONSTANT

| Intercept<br>$E[\beta_i^0]$ | Imple.    | Major<br>Iterations | Function<br>Evaluations | Gradient/Hessian<br>Evaluations | Contraction<br>Iterations |
|-----------------------------|-----------|---------------------|-------------------------|---------------------------------|---------------------------|
| −2                          | NFP-tight | 57                  | 80                      | 58                              | 10,400                    |
|                             | MPEC      | 125                 | 184                     | 126                             |                           |
| −1                          | NFP-tight | 59                  | 82                      | 60                              | 17,100                    |
|                             | MPEC      | 143                 | 274                     | 144                             |                           |
| 0                           | NFP-tight | 55                  | 77                      | 56                              | 29,200                    |
|                             | MPEC      | 112                 | 195                     | 113                             |                           |
| 1                           | NFP-tight | 53                  | 71                      | 54                              | 55,500                    |
|                             | MPEC      | 93                  | 148                     | 94                              |                           |
| 2                           | NFP-tight | 49                  | 68                      | 50                              | 84,000                    |
|                             | MPEC      | 106                 | 188                     | 107                             |                           |
| 3                           | NFP-tight | 48                  | 68                      | 49                              | 146,000                   |
|                             | MPEC      | 84                  | 144                     | 85                              |                           |
| 4                           | NFP-tight | 49                  | 81                      | 50                              | 262,000                   |
|                             | MPEC      | 99                  | 158                     | 100                             |                           |

TABLE V  
MONTE CARLO RESULTS VARYING THE NUMBER OF MARKETS, PRODUCTS, AND SIMULATION  
DRAWS<sup>a</sup>

| # Markets<br>$T$ | # Products<br>$J$ | # Draws<br>$n_s$ | Lipsch.<br>Const. | Imple.    | Runs<br>Converged | CPU Time<br>(hours) | Outside<br>Share |
|------------------|-------------------|------------------|-------------------|-----------|-------------------|---------------------|------------------|
| 100              | 25                | 1000             | 0.999             | NFP-tight | 80%               | 10.9                | 0.45             |
|                  |                   |                  |                   | MPEC      | 100%              | 0.3                 |                  |
| 250              | 25                | 1000             | 0.997             | NFP-tight | 100%              | 22.3                | 0.71             |
|                  |                   |                  |                   | MPEC      | 100%              | 1.2                 |                  |
| 500              | 25                | 1000             | 0.998             | NFP-tight | 80%               | 65.6                | 0.65             |
|                  |                   |                  |                   | MPEC      | 100%              | 2.5                 |                  |
| 100              | 25                | 3000             | 0.999             | NFP-tight | 80%               | 42.3                | 0.46             |
|                  |                   |                  |                   | MPEC      | 100%              | 1                   |                  |
| 250              | 25                | 3000             | 0.997             | NFP-tight | 100%              | 80                  | 0.71             |
|                  |                   |                  |                   | MPEC      | 100%              | 3                   |                  |
| 25               | 100               | 1000             | 0.993             | NFP-tight | 100%              | 5.7                 | 0.28             |
|                  |                   |                  |                   | MPEC      | 100%              | 0.5                 |                  |
| 25               | 250               | 1000             | 0.999             | NFP-tight | 100%              | 28.4                | 0.07             |
|                  |                   |                  |                   | MPEC      | 100%              | 2.3                 |                  |

<sup>a</sup>There is one data set and five starting values for each experiment. The mean intercept is 2. MPEC and NFP produce the same lowest objective value. See the footnote to Table III for other details.

TABLE VI  
ADDITIONAL RESULTS FOR VARYING THE NUMBER OF MARKETS, PRODUCTS, AND  
SIMULATION DRAWS

| # Markets<br>$T$ | # Products<br>$J$ | # Draws<br>$n_s$ | Imple.    | Major<br>Iterations | Function<br>Evaluations | Gradient/Hessian<br>Evaluations | Contraction<br>Iterations |
|------------------|-------------------|------------------|-----------|---------------------|-------------------------|---------------------------------|---------------------------|
| 100              | 25                | 1000             | NFP-tight | 68                  | 130                     | 69                              | 372,248                   |
|                  |                   |                  | MPEC      | 84                  | 98                      | 85                              |                           |
| 250              | 25                | 1000             | NFP-tight | 58                  | 82                      | 59                              | 246,000                   |
|                  |                   |                  | MPEC      | 118                 | 172                     | 119                             |                           |
| 500              | 25                | 1000             | NFP tight | 52                  | 99                      | 53                              | 280,980                   |
|                  |                   |                  | MPEC      | 123                 | 195                     | 124                             |                           |
| 100              | 25                | 3000             | NFP-tight | 60                  | 171                     | 61                              | 479,578                   |
|                  |                   |                  | MPEC      | 83                  | 114                     | 84                              |                           |
| 250              | 25                | 3000             | NFP-tight | 55                  | 68                      | 56                              | 204,000                   |
|                  |                   |                  | MPEC      | 102                 | 135                     | 103                             |                           |
| 25               | 100               | 1000             | NFP-tight | 54                  | 71                      | 55                              | 198,114                   |
|                  |                   |                  | MPEC      | 97                  | 145                     | 98                              |                           |
| 25               | 250               | 1000             | NFP-tight | 60                  | 126                     | 61                              | 359,741                   |
|                  |                   |                  | MPEC      | 75                  | 103                     | 76                              |                           |

We run a final set of Monte Carlo experiments to assess the limits of MPEC’s speed advantage for dense problems. Recall that the sparsity of the problem arises from the fact that there is no interdependence between inverting the econometric error terms across markets. Thus, an MPEC problem with a very large number of markets will have a very sparse Hessian and constraint Jacobian. For relatively sparse problems, we have already shown above that MPEC performs very well even with a large-dimensional parameter vector (e.g., with a large number of  $\xi_{j,t}$ ). We now focus on a dense problem with only  $T = 2$  markets. We then vary the number of products,  $J$ , and hence the number of demand shocks,  $\xi_{j,t}$ . Results are reported in Table VII. For small problems with 250 or fewer products, we find that NFP and MPEC have comparable CPU times, although MPEC maintains its speed advantage for the smallest cases. However, once we consider very large problems with 500 or more products, NFP is found to be faster than MPEC. Therefore, for settings like BLP’s (1995) automobile data with a large number of products and very few markets, MPEC’s speed advantages can be offset in large-dimensional problems for which we cannot exploit the sparsity structure. Whether this will limit the attractiveness of MPEC for actual empirical applications is unclear. At least for the specific Monte Carlo setting we study, the number of products we would need to consider before MPEC loses its speed advantage is considerably larger than what is typically studied in actual empirical settings (e.g., BLP have only around 100 products in their automobile data in any given market).

TABLE VII  
MONTE CARLO RESULTS FOR DENSE PROBLEMS WITH TWO MARKETS AND MANY PRODUCTS

| # Products<br>$J$ | Imple.    | Runs<br>Converged | CPU Time<br>(hours) |
|-------------------|-----------|-------------------|---------------------|
| 25                | NFP-tight | 100%              | 0.63                |
|                   | MPEC      | 100%              | 0.51                |
| 100               | NFP-tight | 100%              | 1.99                |
|                   | MPEC      | 100%              | 1.55                |
| 250               | NFP-tight | 100%              | 12.59               |
|                   | MPEC      | 100%              | 11.77               |
| 500               | NFP-tight | 100%              | 104.8               |
|                   | MPEC      | 100%              | 119.56              |
| 750               | NFP-tight | 100%              | 266.93              |
|                   | MPEC      | 100%              | 433.13              |
| 1000              | NFP-tight | 100%              | 457.96              |
|                   | MPEC      | 100%              | 660.39              |

In the Supplemental Material, we include results from additional Monte Carlo experiments in which the speed advantage of MPEC is shown to be robust to manipulations of the power of the instruments.

### 8. EXTENSION: DYNAMIC DEMAND MODELS

An even more promising frontier for MPEC lies in the application of dynamic demand estimation. Starting with [Melnikov \(2001\)](#), a new stream of literature has considered dynamic analogs of BLP with forward-looking consumers making discrete-choice purchases of durable goods ([Dubé, Hitsch, and Chintagunta \(2010\)](#), [Gordon \(2009\)](#), [Gowrisankaran and Rysman \(2011\)](#), [Lee \(2010\)](#), [Nair \(2007\)](#), [Schiraldi \(2011\)](#)). The typical implementation involves a nested fixed point approach with two nested inner loops. The first inner-loop is the usual numerical inversion of the demand system to obtain the demand shocks,  $\xi$ . The second inner-loop is the iteration of the Bellman equation to obtain the consumers' value functions. In this section, we describe how MPEC can once again serve as a computationally more attractive approach than NFP.

#### 8.1. *Dynamic BLP Model and Algorithms*

We specify a simple model of discrete choice demand for two competing, durable goods with falling prices over time. There is a mass  $M$  of potential consumers at date  $t = 1$ . Consumers drop out of the market once they make a purchase. Abstracting from supply-side specifics, prices evolve over time as a function of the lagged prices of both firms:

$$(10) \quad p_{j,t} = \rho_{j,0} + \rho_{j,1}p_{j,t-1} + \rho_{j,2}p_{-j,t-1} + \psi_{j,t} = p'_{t-1}\rho_j + \psi_{j,t}, \quad j = 1, \dots, 2,$$

where  $\psi_{j,t}$  is a random supply shock. This supply shock is jointly distributed with the demand shock,  $(\xi_{j,t}, \psi_{j,t}) \sim N(0, \Omega)$ , and is independent across time periods, firms, and markets.

A consumer with tastes indexed by  $r = 1, \dots, R$  derives the following flow utility from adopting product  $j = 1, \dots, J$  at time  $t$ :

$$u_{j,t}(p_t; \theta) = \beta_j^r - \alpha^r p_{j,t} + \xi_{j,t} + \epsilon_{j,t}^r,$$

where  $\theta = (\beta_1^r, \dots, \beta_J^r, \alpha^r)$  are taste parameters,  $\xi_{j,t}$  is a time-varying product characteristic, and  $\epsilon_{j,t}^r$  is an independent and identically distributed (i.i.d.) Type I extreme value utility shock. We normalize the expected flow utility from no-purchase to be zero. We assume consumers are forward looking and have rational expectations in the sense that their beliefs about future prices coincide with (10). A consumer  $r$ 's expected value of delaying adoption at time  $t$  is

$$\begin{aligned} (11) \quad v_0^r(p_t; \theta^r) &= \delta \int \left( \log \left( \exp(v_0^r(p'_t \rho_j + \psi; \theta^r)) \right. \right. \\ &\quad \left. \left. + \sum_j \exp(\beta_j^r - \alpha^r(p'_t \rho_j + \psi) + \xi_j) \right) \right) dF_{\psi, \xi}(\psi, \xi), \end{aligned}$$

which satisfies this version of Bellman's equation.<sup>15</sup> The parameter  $\delta \in (0, 1)$  is a discount factor.

We use a discrete distribution with  $R$  mass points to characterize the consumer population's tastes at date  $t = 1$ ,

$$\theta = \begin{cases} \theta^1, & \Pr(1) = \lambda_1, \\ \vdots & \vdots \\ \theta^R, & \Pr(R) = 1 - \sum_{r=1}^{R-1} \lambda_r, \end{cases}$$

where  $\theta^r = (\alpha^r, \beta^r)$ . This heterogeneity allows for certain types of consumers to systematically purchase earlier than others. The remaining mass of consumers of type  $r$  who have not yet adopted at the beginning of period  $t$ ,  $M_t^r$ , is

$$M_t^r = \begin{cases} M \lambda_r, & t = 1, \\ M_{t-1}^r S_0^r(X_{t-1}; \theta^r), & t > 1, \end{cases}$$

<sup>15</sup>We normalize the location parameter of the Type I extreme value distribution to  $-0.577$ .



where  $S_0^r(X_{t-1}; \theta^r)$  is the fraction of consumers of type  $r$  who purchase the outside good. In a given period  $t$ , the market share of product  $j$  is

$$(12) \quad s_j(p_t; \theta) = \sum_{r=1}^R \lambda_{t,r} \frac{\exp(\beta_j^r - \alpha^r p_{j,t} + \xi_{j,t})}{\exp(v_0^r(p_t; \theta^r)) + \sum_{k=1}^J \exp(\beta_k^r - \alpha^r p_{k,t} + \xi_{k,t})},$$

$$j = 1, \dots, J,$$

where

$$\lambda_{t,r} = \begin{cases} \lambda_r, & t = 1, \\ \frac{M_t^r}{\sum_r M_t^r}, & t > 1, \end{cases}$$

is the remaining probability mass associated with type  $r$  consumers at date  $t$ . The assumption of finite types eases dynamic programming because there is only one unknown value-of-waiting function for each type.

The empirical model consists of the system (10) and (12),

$$u_t \equiv \begin{bmatrix} \psi_t \\ \xi_t \end{bmatrix} = \begin{bmatrix} p_t - p'_{t-1} \rho \\ s^{-1}(p_t, S_t; \theta) \end{bmatrix}.$$

The multivariate normal distribution of  $u_t$  induces the following density on the observable outcomes,  $Y_t = (p, S_t)$ :

$$f_Y(Y_t; \theta, \rho, \Omega) = \frac{1}{(2\pi)^J |\Omega|^{1/2}} \exp\left(-\frac{1}{2} u_t' \Omega^{-1} u_t\right) |J_{t,u \rightarrow Y}|,$$

where  $J_{t,u \rightarrow Y}$  is the  $(2J \times 2J)$  Jacobian matrix corresponding to the transformation-of-variables from  $u_t$  to  $Y_t$ . We provide the derivation of the Jacobian in Appendix D.

An NFP approach to maximum likelihood estimation of the model parameters consists of solving the optimization problem

$$(13) \quad \max_{\{\theta, \rho, \Omega\}} \prod_{t=1}^T f_Y(Y_t; \theta, \rho, \Omega).$$

This problem nests two inner loops. Each stage of the outer-loop maximization of the likelihood function in (13) nests a call to compute the fixed point of the Bellman equations, (11), so as to obtain the expected value of delaying adoption. There is also a nested call to compute the demand shocks  $\xi_t$  as the

fixed point of the BLP contraction mapping, (5). Numerical error from both of these inner loops propagates into the outer loop. Thus, the numerical concerns regarding inner-loop convergence tolerance discussed for static BLP may be exacerbated with dynamic analogs of BLP.

Let  $D$  be the support of the state variables. An MPEC approach to maximum likelihood estimation of the model parameters amounts to solving the optimization problem

$$\begin{aligned} \max_{\{\theta, \rho, \Omega, \xi, v\}} \quad & \prod_{t=1}^T \frac{1}{(2\pi)^J |\Omega|^{1/2}} \exp\left(-\frac{1}{2} u_t' \Omega_u^{-1} u_t\right) |J_{t, u \rightarrow Y}| \\ \text{subject to} \quad & s_t(\xi; \theta) = S_t \quad \forall t = 1, \dots, T \quad \text{and} \\ & v_0^r(p_d) \\ & = \delta \log\left(\exp(v_0^r(p_d' \rho_j + \psi)) + \dots\right. \\ & \quad \left. + \sum_j \exp(\beta_j^r - \alpha^r(p_d' \rho_j + \psi) + \xi_j)\right) dF_{\psi, \xi}(\psi, \xi) \\ & \forall d \in D, r = 1, \dots, R. \end{aligned}$$

In this formulation, we now optimize over the demand shocks,  $\xi$ , and the expected value of waiting evaluated at each point,  $v^r(p_d)$ . In this case,  $D \subset \mathbb{R}_+^2$  is the support of the two products' prices. While this approach increases the number of parameters in the outer-loop optimization problem substantially compared to NFP, MPEC eliminates the two inner loops. Chebyshev approximation of  $v_0^r(p; \theta^r)$  (Judd (1998)) reduces the dimension of this problem by searching over the Chebyshev weights, rather than over the value function at each point in a discretized state space. For details, see Appendix C.

## 8.2. Dynamic BLP Monte Carlo Experiments

We construct several Monte Carlo experiments to assess the relative performance of MPEC versus NFP in the context of the dynamic demand model. Details about the data-generating process are provided in Appendix B. It is straightforward to show that the speed of the fixed point calculation associated with the consumer's expected value of waiting is related to the discount factor. Therefore, we compare performance with two different discount factors,  $\delta = 0.96$  and  $\delta = 0.99$ , corresponding to the annual rate and the quarterly rate, respectively, for an annual interest rate of 4%.

It is difficult to derive closed-form expressions for the Jacobian of the outer-loop optimization associated with dynamic BLP, under both NFP and MPEC.

However, due to its formulation as a standard constrained-optimization problem, the MPEC algorithm can exploit automatic differentiation to obtain exact derivatives for the outer loop (Griewank and Corliss (1992)).<sup>16</sup> Supplying automatic derivatives to the NFP formulation will slow the algorithm considerably due to the large number of calls to the value function during the inner loop.<sup>17</sup> In practice, this presents the researcher using NFP with a difficult trade-off between speed and numerical accuracy. In our results below, we compare NFP using automatic and numerical derivatives.

Results from the Monte Carlo experiments using  $\delta = 0.99$  and  $\delta = 0.96$  are presented in Tables VIII and IX, respectively. As expected, with tight inner- and outer-loop tolerances, each algorithm produces very similar estimates. However, on average, MPEC is considerably faster than NFP. With  $\delta = 0.99$ , MPEC is twice as fast as NFP with a numerical gradient and more than ten times as fast as NFP with an analytic gradient. With five starting values, a typical run of MPEC requires 0.89 hours, whereas NFP requires 11.0 hours with an automatic gradient and 1.83 hours with a numerical gradient. Given our findings for the static BLP estimator, we expect the relative speed of MPEC to increase considerably as automatic differentiation algorithms improve. A note of caution is in order for NFP with a numerical gradient. Even though it produced comparable estimates of the structural parameters, the algorithm failed to achieve convergence in the majority of replications. The lack of reliability in convergence for NFP with a numerical gradient poses a problem to the researcher who would not know the true parameter values in practice.

Our results are quite comparable using  $\delta = 0.96$ , with MPEC still emerging as the fastest algorithm. As expected, the speed of NFP with numerical derivatives improves in this case because the lower discount factor speeds the convergence of the Bellman equation iteration. As before, the NFP algorithm fails to diagnose convergence to a local optimum in the majority of replications when a numerical gradient is used. We also compared NFP and MPEC for the homogeneous logit with  $R = 1$ . Because the market shares can be inverted analytically, this case isolates the role of the iteration of the Bellman equation. Here, too, we find a large relative speed advantage for MPEC. Results for the  $R = 1$  case are available in the Supplemental Material.

## 9. CONCLUSIONS

In this paper, we analyze the numerical properties of the NFP approach proposed by BLP to estimate the random coefficients logit demand model. In practice, inverting the market share equations in NFP's inner-loop using contraction mapping iterations may be time consuming. Consequently, researchers

<sup>16</sup>We use the MAD (MATLAB Automatic Differentiation) package, which is part of TOMLAB.

<sup>17</sup>It is important to impose tight inner-loop convergence of both the levels and gradients.

TABLE VIII  
MONTE CARLO RESULTS FOR DYNAMIC BLP WITH TWO CONSUMER TYPES AND  $\delta = 0.99^a$

|  |        | MPEC       |        | NFP Auto.  |        | NFP numerical |        |
|--|--------|------------|--------|------------|--------|---------------|--------|
| CPU Time (hours):  |        | 0.8964     |        | 10.9582    |        | 1.8306        |        |
| Parameters   | TRUE   | Mean       | RMSE   | Mean       | RMSE   | Mean          | RMSE   |
| Utility intercept, product 1                             | 4      | 4.0648     | 0.6967 | 4.0648     | 0.6967 | 4.0647        | 0.6966 |
| Utility intercept, product 2                             | 3      | 3.0904     | 0.6798 | 3.0904     | 0.6798 | 3.0902        | 0.6797 |
| Utility price coefficient, type 1                        | -1     | -1.0049    | 0.0215 | -1.0049    | 0.0215 | -1.0049       | 0.0215 |
| Utility price coefficient, type 2                        | -2     | -2.4481    | 1.3784 | -2.4481    | 1.3784 | -2.4481       | 1.3784 |
| Frequency, type 1  | 0.7    | 0.7008     | 0.0724 | 0.5580     | 0.2497 | 0.6660        | 0.1383 |
| Price, product 1, constant                               | 0.2    | 0.1915     | 0.0710 | 0.1915     | 0.0710 | 0.1915        | 0.0710 |
| Price, product 1, lagged price of product 1              | 0.8    | 0.7876     | 0.0407 | 0.7876     | 0.0407 | 0.7876        | 0.0407 |
| Price, product 1, lagged price of product 2              | 0      | 0.0060     | 0.0379 | 0.0060     | 0.0379 | 0.0060        | 0.0379 |
| Price, product 2, constant                               | 0.2    | 0.2300     | 0.0949 | 0.2300     | 0.0949 | 0.2300        | 0.0949 |
| Price, product 2, lagged price of product 1              | 0      | 0.0010     | 0.0297 | 0.0010     | 0.0297 | 0.0010        | 0.0297 |
| Price, product 2, lagged price of product 2              | 0.8    | 0.7907     | 0.0345 | 0.7907     | 0.0345 | 0.7907        | 0.0345 |
| Demand shocks, Cholesky variance term                    | 1      | 0.9952     | 0.0226 | 0.9952     | 0.0226 | 0.9952        | 0.0226 |
| Supply shocks, Cholesky variance term                    | 0.8660 | 0.8605     | 0.3615 | 0.8605     | 0.3615 | 0.8605        | 0.3615 |
| Covariance btw supply and demand, Cholesky variance term | 0.5    | 0.5048     | 0.3637 | 0.5048     | 0.3637 | 0.5048        | 0.3637 |
| Mean of obj function at solution                         |        | -1269.4908 |        | -1269.4908 |        | -1269.4908    |        |
| % of replications routine reports convergence            |        | 100        |        | 100        |        | 20            |        |

<sup>a</sup>There are 20 replications. Each replication uses five starting values to do a better job at finding a global minimum. There is no inner loop in MPEC;  $\epsilon_{\text{out}} = 10^{-6}$  and  $\epsilon_{\text{feasible}} = 10^{-6}$ . For NFP, we use the tolerance  $\epsilon = 10^{-14}$  for both inner loops. The data have  $T = 50$  periods and  $M = 5$  distinct markets. Each market has two competing products. The Chebyshev regression approximation to the value function uses a fourth-order polynomial and four interpolation nodes. The numerical integration of future states uses Gauss–Hermite quadrature with three nodes. The code uses TOMLAB’s MAD package for automatic differentiation.

TABLE IX  
MONTE CARLO RESULTS FOR DYNAMIC BLP WITH TWO CONSUMER TYPES AND  $\delta = 0.96^a$

|  |        | MPEC      |        | NFP MAD   |        | NFP numerical |        |
|--|--------|-----------|--------|-----------|--------|---------------|--------|
| CPU Time (hours):  |        | 0.7978    |        | 11.4687   |        | 1.3712        |        |
| Parameters   | TRUE   | Mean      | RMSE   | Mean      | RMSE   | Mean          | RMSE   |
| Utility intercept, product 1                             | 4      | 4.0203    | 0.4393 | 4.0203    | 0.4392 | 4.0203        | 0.4392 |
| Utility intercept, product 2                             | 3      | 3.0453    | 0.4185 | 3.0453    | 0.4185 | 3.0453        | 0.4185 |
| Utility price coefficient, type 1                        | -1     | -1.0029   | 0.0247 | -1.0029   | 0.0247 | -1.0029       | 0.0247 |
| Utility price coefficient, type 2                        | -2     | -2.0549   | 0.2742 | -2.0549   | 0.2742 | -2.0549       | 0.2742 |
| Frequency, type 1  | 0.7    | 0.6985    | 0.0689 | 0.5079    | 0.2845 | 0.6985        | 0.0689 |
| Price, product 1, constant                               | 0.2    | 0.1926    | 0.0707 | 0.1926    | 0.0707 | 0.1926        | 0.0707 |
| Price, product 1, lagged price of product 1              | 0.8    | 0.7887    | 0.0354 | 0.7887    | 0.0354 | 0.7887        | 0.0354 |
| Price, product 1, lagged price of product 2              | 0      | 0.0043    | 0.0332 | 0.0043    | 0.0332 | 0.0043        | 0.0332 |
| Price, product 2, constant                               | 0.2    | 0.2314    | 0.0963 | 0.2314    | 0.0963 | 0.2315        | 0.0963 |
| Price, product 2, lagged price of product 1              | 0      | 0.0016    | 0.0292 | 0.0016    | 0.0292 | 0.0016        | 0.0292 |
| Price, product 2, lagged price of product 2              | 0.8    | 0.7896    | 0.0354 | 0.7896    | 0.0354 | 0.7896        | 0.0354 |
| Demand shocks, Cholesky variance term                    | 1      | 0.9959    | 0.0235 | 0.9959    | 0.0235 | 0.9959        | 0.0235 |
| Supply shocks, Cholesky variance term                    | 0.8660 | 0.8602    | 0.3612 | 0.8602    | 0.3612 | 0.8602        | 0.3612 |
| Covariance btw supply and demand, Cholesky variance term | 0.5    | 0.5039    | 0.3645 | 0.5039    | 0.3645 | 0.5039        | 0.3645 |
| Mean of obj function at solution                         |        | -694.5832 |        | -694.5832 |        | -694.5832     |        |
| % of replications routine reports convergence            |        | 100       |        | 100       |        | 40            |        |

<sup>a</sup>There are 20 replications. Each replication uses five starting values to do a better job at finding a global minimum. There is no inner loop in MPEC;  $\epsilon_{\text{out}} = 10^{-6}$  and  $\epsilon_{\text{feasible}} = 10^{-6}$ . For NFP, we use the tolerance  $\epsilon = 10^{-14}$  for both inner loops. The data have  $T = 50$  periods and  $M = 5$  distinct markets. Each market has two competing products. The Chebyshev regression approximation to the value function uses a fourth-order polynomial and four interpolation nodes. The numerical integration of future states uses Gauss–Hermite quadrature with three nodes. The code uses TOMLAB’s MAD package for automatic differentiation.

may be tempted to use a loose inner-loop stopping criterion. Using numerical theory and computational examples with both pseudo-real and synthetic data, we demonstrate that setting loose inner-loop tolerances can lead to incorrect parameter estimates or a failure of an optimization routine to report convergence.

We propose a new constrained optimization formulation, MPEC, which avoids the inner loop for repeatedly inverting the market share equations and, hence, eliminates the numerical error in evaluating the objective function and its gradient. MPEC produces good estimates relatively quickly for most of the data-generating processes that we consider. Its speed is invariant to the Lipschitz constant of the inner-loop contraction mapping used in NFP, as expected. In the case of a very dense, large-dimensional problem with few markets and many products, we lose MPEC's speed advantage over NFP. A caveat to our findings is that we have focused on the standard implementation of BLP's logit demand system. We cannot rule out that our relative speed findings for MPEC versus NFP could change for more complex demand systems.

We adapt the MPEC approach to a new class of applications with forward-looking consumers. The relative advantage of MPEC remains strong with dynamics because two inner loops must be solved: the dynamic programming problem and the market share inversion. This burdensome collection of three loops (optimization, market shares, dynamic programming) makes the traditional BLP approach nearly untenable in terms of computational time. Automatic differentiation is extremely slow with NFP due to the inner-loop. Alternatively, using numerical derivatives with NFP can produce convergence problems with the outer loop. Current work (Lee (2010)) further extends the number of inner loops being solved in estimation. As demand models become richer, we conjecture that the computational benefits of MPEC over NFP will become greater.

While we have conducted our analysis in the context of demand estimation, our numerical theory results, along with several of our insights, can be extended to broader empirical contexts using a nested fixed point approach. In future research, we see strong potential for MPEC in the context of demand models where there is a unique vector of demand shocks that rationalize the market shares, but no contraction mapping (e.g., Berry, Gandhi, and Haile (2011)). For example, one can theoretically recast the estimator of Berry and Pakes (2007) as a mathematical programming problem with complementarity constraints (Luo, Pang, and Ralph (1996), Leyffer, Lopez-Calva, and Nocedal (2006), Leyffer and Munson (2009)). While the dimension of the problem remains beyond the capabilities of current solvers, we expect this example to be an interesting future extension of our work.

## APPENDIX A: PROOFS

PROOF OF LEMMA 1: By a Taylor series expansion around  $\xi(\theta, 0)$ , we have

$$\begin{aligned} & Q(\xi(\theta, \epsilon_{\text{in}})) - Q(\xi(\theta, 0)) \\ &= \left[ \frac{\partial Q(\xi(\theta))}{\partial \xi} \bigg|_{\xi=\xi(\theta, 0)} \right]' (\xi(\theta, \epsilon_{\text{in}}) - \xi(\theta, 0)) \\ &+ O(\|\xi(\theta, \epsilon_{\text{in}}) - \xi(\theta, 0)\|^2) \end{aligned}$$

and

$$\begin{aligned} & \nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\theta, \epsilon_{\text{in}})} - \nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\theta, 0)} \\ &= \left[ \frac{\partial \nabla_{\theta} Q(\xi(\theta))}{\partial \xi} \bigg|_{\xi=\xi(\theta, 0)} \right]' (\xi(\theta, \epsilon_{\text{in}}) - \xi(\theta, 0)) \\ &+ O(\|\xi(\theta, \epsilon_{\text{in}}) - \xi(\theta, 0)\|^2). \end{aligned}$$

Because  $\|\xi(\theta, \epsilon_{\text{in}}) - \xi(\theta, 0)\| \leq \frac{L(\theta)}{1-L(\theta)} \epsilon_{\text{in}}$  by the contraction mapping theorem, and assuming both

$$\left\| \frac{\partial Q(\xi(\theta))}{\partial \xi} \bigg|_{\xi=\xi(\theta, 0)} \right\| \quad \text{and} \quad \left\| \frac{\partial \nabla_{\theta} Q(\xi(\theta))}{\partial \xi} \bigg|_{\xi=\xi(\theta, 0)} \right\|$$

are bounded, we obtain

$$\begin{aligned} |Q(\xi(\theta, \epsilon_{\text{in}})) - Q(\xi(\theta, 0))| &= O\left(\frac{L(\theta)}{1-L(\theta)} \epsilon_{\text{in}}\right), \\ \|\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\theta, \epsilon_{\text{in}})} - \nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\theta, 0)}\| \\ &= O\left(\frac{L(\theta)}{1-L(\theta)} \epsilon_{\text{in}}\right). \end{aligned} \quad Q.E.D.$$

PROOF OF THEOREM 2: Because  $\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})} = 0$ , the second result in Lemma 1 gives

$$(14) \quad \|\nabla_{\theta} Q(\xi(\theta))|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)}\| = O\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1-L(\hat{\theta}(\epsilon_{\text{in}}))} \epsilon_{\text{in}}\right).$$

Note that we evaluate the gradient of the GMM objective function with no numerical error at  $\hat{\theta}(\epsilon_{\text{in}})$ .

Let  $\bar{\theta}$  be any value of the structural parameters near  $\hat{\theta}(\epsilon_{\text{in}})$ . By first the inverse triangle inequality, then the regular triangle inequality, and then finally



a Taylor series expansion, we have

$$\begin{aligned}
 & \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})} \right\| - \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| \\
 & \leq \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})} - \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| \\
 & = \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})} - \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, 0)} \right. \\
 & \quad \left. + \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, 0)} - \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| \\
 & \leq \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})} - \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, 0)} \right\| \\
 & \quad + \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, 0)} - \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| \\
 & \leq O\left(\frac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\epsilon_{\text{in}}\right) + \left\| \nabla_{\theta}^2 Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| \|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\| \\
 & \quad + O(\|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\|^2).
 \end{aligned}$$

Assuming  $\|\nabla_{\theta}^2 Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)}\|$  is bounded, the second-order term  $O(\|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\|^2)$  can be ignored. By rearranging the above inequality, we obtain

$$\begin{aligned}
 & \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\tilde{\theta}, \epsilon_{\text{in}})} \right\| \\
 & \leq \left\| \nabla_{\theta} Q(\xi(\theta)) \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| + O\left(\frac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\epsilon_{\text{in}}\right) + O(\|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\|) \\
 & = O\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1-L(\hat{\theta}(\epsilon_{\text{in}}))}\epsilon_{\text{in}}\right) + O\left(\frac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\epsilon_{\text{in}}\right) + O(\|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\|) \\
 & = O\left(\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1-L(\hat{\theta}(\epsilon_{\text{in}}))} + \frac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\right)\epsilon_{\text{in}} + \|\tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}})\|\right). \quad Q.E.D.
 \end{aligned}$$

**PROOF OF THEOREM 3:** First, we can quantify the bias between  $Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}))$  and  $Q(\xi(\theta^*, 0))$ :

$$\begin{aligned}
 & Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\theta^*, 0)) \\
 & = Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)) \\
 & \quad + Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)) - Q(\xi(\theta^*, 0)) \\
 & = \left[ \frac{\partial Q(\xi(\theta))}{\partial \xi} \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right]' (\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)) \\
 & \quad + O(\|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\|^2)
 \end{aligned}$$

$$\begin{aligned}
& + [\nabla_{\theta} Q(\xi(\theta^*))]'(\hat{\theta}(\epsilon_{\text{in}}) - \theta^*) + O(\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\|^2) \\
& = \left[ \frac{\partial Q(\xi(\theta))}{\partial \xi} \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right]' (\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)) \\
& \quad + O(\|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\|^2) + O(\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\|^2),
\end{aligned}$$

because  $\nabla_{\theta} Q(\xi(\theta^*)) = 0$  at the true estimates  $\theta^*$ .

Rearranging the equality involving  $Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\theta^*, 0))$  to focus on the  $O(\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\|^2)$  term, we have

$$\begin{aligned}
& O(\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\|^2) \\
& = Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\theta^*, 0)) \\
& \quad - \left[ \frac{\partial Q(\xi(\theta))}{\partial \xi} \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right]' (\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)) \\
& \quad - O(\|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\|^2) \\
& \leq |Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\theta^*, 0))| \\
& \quad + \left\| \frac{\partial Q(\xi(\theta))}{\partial \xi} \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\| \|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\| \\
& \quad - O(\|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\|^2).
\end{aligned}$$

Assuming  $\left\| \frac{\partial Q(\xi(\theta))}{\partial \xi} \Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}), 0)} \right\|$  is bounded, the second-order term  $O(\|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\|^2)$  can be ignored. From the contraction mapping theorem, we know

$$\|\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}) - \xi(\hat{\theta}(\epsilon_{\text{in}}), 0)\| \leq \frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1 - L(\hat{\theta}(\epsilon_{\text{in}}))} \epsilon_{\text{in}}.$$

Hence, we obtain

$$\begin{aligned}
O(\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\|^2) & \leq |Q(\xi(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}})) - Q(\xi(\theta^*, 0))| \\
& \quad + O\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1 - L(\hat{\theta}(\epsilon_{\text{in}}))} \epsilon_{\text{in}}\right). \quad Q.E.D.
\end{aligned}$$

**PROOF OF PROPOSITION 4:** The first-order condition of the GMM estimation problem (4) is

$$(15) \quad \frac{dQ(\xi(\theta))}{d\theta} = \frac{d\xi(\theta)'}{d\theta} \frac{\partial Q}{\partial \xi} = 0.$$

The Lagrangian of the constrained optimization problem (8) is  $\mathcal{L}(\theta, \xi, \lambda) = Q(\xi) + \lambda'(S - s(\xi; \theta))$ , where  $\lambda$  is the vector of Lagrange multipliers. The first-order conditions of (8) are

$$(16) \quad \begin{aligned} \frac{\partial \mathcal{L}(\theta, \xi, \lambda)}{\partial \theta} &= -\frac{\partial s(\xi; \theta)'}{\partial \theta} \lambda = 0, \\ \frac{\partial \mathcal{L}(\theta, \xi, \lambda)}{\partial \xi} &= \frac{\partial Q(\xi)}{\partial \xi} - \frac{\partial s(\xi; \theta)'}{\partial \xi} \lambda = 0, \\ \frac{\partial \mathcal{L}(\theta, \xi, \lambda)}{\partial \lambda} &= S - s(\xi; \theta) = 0. \end{aligned}$$

First, we prove that the matrix  $\frac{\partial s(\xi; \theta)'}{\partial \xi}$  evaluated at  $(\theta^*, \xi^*)$  is nonsingular.<sup>18</sup> Recall that the Lipschitz constant of the contraction mapping (5) is defined as  $L(\theta) = \max_{\xi \in \Xi} \|I - \nabla_{\xi}(\log s(\xi; \theta))\|_{\infty}$ . Consequently,

$$\|I - \nabla_{\xi}(\log s(\xi^*; \theta^*))\|_{\infty} \leq L(\theta^*) < 1.$$

To simplify the notation, denote  $A = \nabla_{\xi}(\log s(\xi^*; \theta^*))$ . We claim that  $A$  is nonsingular. Suppose  $A$  is singular so that a  $v \neq 0$  solves  $Av = 0$ . It follows that

$$\begin{aligned} 0 &= \|Av\|_{\infty} = \|v - (I - A)v\|_{\infty} \geq \|v\|_{\infty} - \|(I - A)v\|_{\infty} \\ &\geq \|v\|_{\infty} - \|I - A\|_{\infty} \|v\|_{\infty} > 0, \end{aligned}$$

which gives a contraction. Because  $\frac{\partial s(\xi^*; \theta^*)}{\partial \xi} = (\text{diag}(S)) * A$ , the matrix  $\frac{\partial s(\xi^*; \theta^*)'}{\partial \xi}$  is nonsingular.

Solving the second set of first-order conditions for  $\lambda$  gives  $\lambda^* = \left(\frac{\partial s(\xi^*; \theta^*)'}{\partial \xi}\right)^{-1} \times \frac{\partial Q(\xi^*)}{\partial \xi}$ . Then, by the implicit function theorem, we have

$$\begin{aligned} 0 &= \frac{\partial \mathcal{L}}{\partial \theta} = -\frac{\partial s(\xi^*; \theta^*)'}{\partial \theta} \left(\frac{\partial s(\xi^*; \theta^*)'}{\partial \xi}\right)^{-1} \frac{\partial Q(\xi^*)}{\partial \xi} \\ &= \frac{d\xi(\theta^*)'}{d\theta} \frac{\partial Q(\xi^*)}{\partial \xi} = \frac{dQ(\xi(\theta^*))}{d\theta} = 0. \end{aligned} \quad Q.E.D.$$

## APPENDIX B: THE SYNTHETIC DATA-GENERATING PROCESS

### B.1. Static Model

We base our synthetic datasets on the demand model in Section 2. We construct  $T = 50$  independent markets, each with the same set of  $J = 25$  products.

<sup>18</sup>We thank Kenneth Judd and John Birge for pointing out this property. Rust (1988, p. 1016) showed the same result for dynamic discrete-choice models using the Banach inverse theorem.

Each product  $j$  has  $K = 3$  observed, market-invariant characteristics that are generated as

$$\begin{bmatrix} x_{1,j} \\ x_{2,j} \\ x_{3,j} \end{bmatrix} \sim N \left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.8 & 0.3 \\ -0.8 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix} \right).$$

In addition, each product  $j$  has a market-specific vertical characteristic:  $\xi_{j,t} \sim \text{i.i.d. } N(0, 1)$ . Finally, each product  $j$  has a market-specific price generated as

$$p_{j,t} = \left| 0.5 \cdot \xi_{j,t} + e_{j,t} + 1.1 \cdot \sum_{k=1}^3 x_{k,j} \right|,$$

where  $e_{j,t} \sim N(0, 1)$  is an innovation that enters price. For the Monte Carlo experiments in Section 7, we change the pricing process as follows:  $p_{j,t} = 3 + \xi_{j,t} \cdot 1.5 + u_{j,t} + \sum_{k=1}^3 x_{k,j,t}$ , where  $u_{j,t}$  is a uniform(0, 5) random variable.

For each product  $j$  in market  $t$ , there is a separate vector,  $z_{j,t}$ , of  $D = 6$  underlying instruments generated as follows:  $z_{j,t,d} \sim U(0, 1) + \frac{1}{4}(e_{j,t} + 1.1 \cdot \sum_{k=1}^3 x_{k,j,t})$ , where  $U(0, 1)$  is the realization of a uniform random variable and  $e_{j,t}$  is the price innovation. In addition, we also use higher-order polynomial expansions of the excluded instruments,  $z_{jt}$ , and the exogenous regressors,  $x_j$ :  $z_{j,t,d}^2$ ,  $z_{j,t,d}^3$ ,  $x_{j,k}^2$ ,  $x_{j,k}^3$ ,  $\prod_{d=1}^D z_{j,t,d}$ ,  $\prod_{k=1}^K x_{j,k}$ ,  $z_{j,t,d} \cdot x_{j1}$ , and  $z_{j,t,d} \cdot x_{j,t,2}$ . There are 42 total moments.

There are five dimensions of consumer preference,  $\beta_i = \{\beta_i^0, \beta_i^1, \beta_i^2, \beta_i^3, \beta_i^p\}$  (an intercept,  $K = 3$  attributes, and price), each distributed independently normal with means and variances  $E[\beta_i] = \{-1.0, 1.5, 1.5, 0.5, -3.0\}$  and  $\text{Var}[\beta_i] = \{0.5, 0.5, 0.5, 0.5, 0.2\}$ .

We simulate the integral in the market share equation, (3), with  $n_s = 1000$  independent standard normal draws. Because our focus is not on numerical integration error, we use the same set of 1000 draws to compute market shares in the data-generation and estimation phases.

## B.2. Dynamic Model

For the dynamic model, we allow for a distribution of tastes with two mass points over the sensitivity to prices:

$$(\beta_1, \beta_2, \alpha)' = \begin{cases} (4, 3, -1), & \text{with probability } \lambda = 0.7, \\ (4, 3, -2), & \text{with probability } (1 - \lambda) = 0.3. \end{cases}$$

We assume that prices are generated as follows:

$$\begin{aligned} p_{1,t} &= 5 + 0.8p_{1,t-1} + 0.0p_{2,t-1} + \psi_{1,t}, \\ p_{2,t} &= 5 + 0.0p_{1,t-1} + 0.8p_{2,t-1} + \psi_{2,t}. \end{aligned}$$

Finally, we assume the supply and demand shocks satisfy  $(\psi_{j,t}, \xi_{j,t}) \sim N(0, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix})$  and are independent across markets and time periods. For our Chebyshev approximation of  $v_0^r(p_i; \theta^r)$ , we use six nodes and a fourth-order polynomial. For the NFP algorithm, we use a tolerance of  $10^{-14}$  for both inner loops. We use data on  $M = 5$  independent markets and  $T = 50$  time periods per market.

#### APPENDIX C: CHEBYSHEV APPROXIMATION OF THE EXPECTED VALUE OF WAITING

First, we bound the range of prices as  $p = (p_1, p_2)' \in [0, b] \times [0, b]$ , where  $b$  is large ( $b$  is 1.5 times the largest observed price in the data). We then approximate the expected value of delaying adoption with Chebyshev polynomials,  $v_0^r(p; \theta^r) \approx \gamma^r \Lambda(p)$ , where  $\gamma^r$  is a  $K \times 1$  vector of parameters and  $\Lambda(p)$  is a  $K \times 1$  vector of  $K$  Chebyshev polynomials. Therefore, we can rewrite the Bellman equation as

$$\begin{aligned} \gamma^r \Lambda(p) = & \delta \int \log \left( \exp(\gamma^r \Lambda(pp + \psi)) \right. \\ & \left. + \sum_j \exp(\beta_j^r - \alpha^r(p' \rho_j + \psi) + \xi_j) \right) dF_{\psi, \xi}(\psi, \xi). \end{aligned}$$

To solve for the Chebyshev weights, we use the Galerkin method described in Judd (1992). We define the residual function

$$\begin{aligned} (17) \quad R(p; \gamma) = & \gamma^r \Lambda(p) \\ & - \delta \int \log \left( \exp(\gamma^r \Lambda(pp + \psi)) \right. \\ & \left. + \sum_j \exp(\beta_j^r - \alpha^r(p' \rho_j + \psi) + \xi_j) \right) dF_{\psi, \xi}(\psi, \xi). \end{aligned}$$

Next, we let  $X$  be the matrix of  $K$  Chebyshev polynomials at each of the  $G$  points on our grid (i.e.,  $G$  nodes). Our goal is to search for parameters,  $\gamma$ , that set the following expression to zero:

$$X'R(p; \gamma) = 0.$$

We use an iterated least squares approach for NFP:

- (i) Pick a starting value  $\gamma^{r,0}$ ,  $v_0^{r,0}(p; \Theta^r) = \gamma^{r,0'} \rho(p)$ .

(ii) Use quadrature to compute

$$Y(p; \gamma^{r,0}) = \delta \int \log \left( \exp(\gamma^{r,0'} \Lambda(p\rho + \psi)) \right. \\ \left. + \sum_j \exp(\beta_j^r - \alpha^r(p' \rho_j + \psi) + \xi_j) \right) dF_{\psi, \xi}(\psi, \xi).$$

(iii) Solve the least squares problem:  $\min_{\gamma} R(p; \gamma)' R(p; \gamma)$  or

$$\min_{\gamma} (X\gamma^r - Y(p; \gamma^{r,0}))' (X\gamma^r - Y(p; \gamma^{r,0}))$$

for which the solution is  $\gamma^{r,1} = (X'X)^{-1}X'Y(p; \gamma^{r,0})$ .

(iv) Compute  $v_0^{r,1}(p; \Theta^r) = \gamma^{r,1'} \Lambda(p)$ .

(v) Repeat steps (ii) and (iii) until convergence.

#### APPENDIX D: JACOBIAN OF THE DENSITY OF $(p_t, S_t)$ IN THE DYNAMIC BLP MODEL

The Jacobian is defined as

$$J_{t,u \rightarrow Y} = \begin{bmatrix} \frac{\partial \psi_t}{\partial p_t} & \frac{\partial \psi_t}{\partial S_t} \\ \frac{\partial \xi_t}{\partial p_t} & \frac{\partial \xi_t}{\partial S_t} \end{bmatrix}.$$

Since  $\frac{\partial \psi_t}{\partial \log(p_t)} = I_J$  and  $\frac{\partial \psi_t}{\partial \log(p_t)} = 0_J$  (a square matrix of zeros), we only need to compute the matrix of derivatives,  $[\frac{\partial \xi_t}{\partial S_t}]$ . We can simplify this calculation by applying the implicit function theorem to the system

$$G(S_t, \xi_t) = s(p, \xi_t; \theta) - S_t = 0$$

and computing the lower block of the Jacobian as

$$J_{t, \xi \rightarrow S} = - \left[ \frac{\partial G}{\partial \xi_t} \right]^{-1} \left[ \frac{\partial G}{\partial S_t} \right] = \left[ \frac{\partial s}{\partial \xi_t} \right]^{-1},$$

where the  $(j, k)$  element of  $\frac{\partial s_{j,t}}{\partial \xi_{k,t}}$  is

$$\frac{\partial s_{j,t}}{\partial \xi_{k,t}} = \begin{cases} \sum_r \lambda_{r,t} s_j(p_t, \xi_t; \theta^r) (1 - s_j(p_t, \xi_t; \theta^r)), & \text{if } j = k, \\ - \sum_r \lambda_{r,t} s_j(p, \xi_t; \theta^r) s_k(p, \xi_t; \theta^r), & \text{otherwise.} \end{cases}$$

## REFERENCES

- ACKERBERG, D., J. GEWEKE, AND J. HAHN (2009): "Comments on 'Convergence Properties of the Likelihood of Computed Dynamic Models'," by J. Fernandez-Villaverde, J. F. Rubio-Ramirez, and J. S. Santos, *Econometrica*, 77 (6), 2009–2017. [2241]
- BERRY, S. (1994): "Estimating Discrete-Choice Models of Product Differentiation," *RAND Journal of Economics*, 25 (2), 242–262. [2232,2236]
- BERRY, S., AND P. A. HAILE (2011): "Identification in Differentiated Products Markets Using Market Level Data," Working Paper, Yale University. [2235]
- BERRY, S., AND A. PAKES (2007): "The Pure Characteristics Demand Model," *International Economic Review*, 48 (4), 1193–1225. [2258]
- BERRY, S., A. GANDHI, AND P. HAILE (2011): "Connected Substitutes and Invertibility of Demand," Working Paper, Yale University. [2258]
- BERRY, S., J. LEVINSOHN, AND A. PAKES (1995): "Automobile Prices in Market Equilibrium," *Econometrica*, 63 (4), 841–890. [2231,2238,2246,2250]
- BERRY, S., O. B. LINTON, AND A. PAKES (2004): "Limit Theorems for Estimating the Parameters of Differentiated Product Demand Systems," *Review of Economic Studies*, 71 (3), 613–654. [2232]
- BYRD, R. H., M. E. HRIBAR, AND J. NOCEDAL (1999): "An Interior Point Method for Large Scale Nonlinear Programming," *SIAM Journal on Optimization*, 9 (4), 877–990. [2243]
- BYRD, R. H., J. NOCEDAL, AND R. A. WALTZ (2006): "KNITRO: An Integrated Package for Nonlinear Optimization," in *Large-Scale Nonlinear Optimization*, ed. by G. di Pillo and M. Roma. New York: Springer, 35–59. [2243]
- CARRANZA, J. E. (2010): "Product Innovation and Adoption in Market Equilibrium: The Case of Digital Cameras," *International Journal of Industrial Organization*, 28 (6), 604–618. [2234]
- CHIAPPORI, P.-A., AND I. KOMMUNJER. (2009): "On the Nonparametric Identification of Multiple Choice Models," Working Paper, Columbia University. [2235]
- DAHLQUIST, G., AND Å. BJÖRCK (2008): *Numerical Methods in Scientific Computing*. Philadelphia, PA: SIAM. [2237]
- DUBÉ, J.-P., J. T. FOX, AND C.-L. SU (2012): "Supplement to 'Improving the Numerical Performance of Static and Dynamic Aggregate Discrete Choice Random Coefficients Demand Estimation'," *Econometrica Supplemental Material*, 80, [http://www.econometricsociety.org/ecta/Supmat/8585\\_extensions.pdf](http://www.econometricsociety.org/ecta/Supmat/8585_extensions.pdf); [http://www.econometricsociety.org/ecta/Supmat/8585\\_data\\_and\\_programs.zip](http://www.econometricsociety.org/ecta/Supmat/8585_data_and_programs.zip). [2243]
- DUBÉ, J.-P., G. HITSCH, AND P. CHINTAGUNTA (2010): "Tipping and Concentration in Markets With Indirect Network Effects," *Marketing Science*, 29 (2), 216–249. [2251]
- FOX, J. T., AND A. GANDHI (2011): "Identifying Demand With Multidimensional Unobservables: A Random Functions Approach," Working Paper, University of Michigan. [2235]
- FOX, J. T., K.-I. KIM, S. P. RYAN, AND P. BAJARI (2012): "The Random Coefficients Logit Model Is Identified," *Journal of Econometrics*, 166 (2), 204–212. [2235]
- GORDON, B. (2009): "A Dynamic Model of Consumer Replacement Cycles in the PC Processor Industry," *Marketing Science*, 28 (5), 846–867. [2251]
- GOWRISANKARAN, G., AND M. RYSMAN (2011): "Dynamics of Consumer Demand for New Durable Goods," Working Paper, The University of Arizona. [2234,2251]
- GRIEWANK, A., AND G. F. CORLISS (1992): *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. Philadelphia, PA: SIAM. [2255]
- HENDEL, I., AND A. NEVO (2007): "Measuring the Implications of Sales and Consumer Inventory Behavior," *Econometrica*, 74 (16), 1637–1673. [2234]
- JUDD, K. L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58 (2), 410–452. [2264]
- (1998): *Numerical Methods in Economics*. Cambridge, MA: MIT Press. [2235,2254]
- KNITTEL, C. R., AND K. METAXOGLU (2008): "Estimation of Random Coefficient Demand Models: Challenges, Difficulties and Warnings," Working Paper, MIT. [2246]



- LEE, R. S. (2010): "Vertical Integration and Exclusivity in Platform and Two-Sided Markets," Working Paper, New York University. [2234,2251,2258]
- LEYFFER, S., AND T. MUNSON (2009): "A Globally Convergent Filter Method for MPECs," Preprint ANL/MCS-P1565-1208, Argonne National Laboratory. [2258]
- LEYFFER, S., G. LOPEZ-CALVA, AND J. NOCEDAL (2006): "Interior Methods for Mathematical Programs With Complementarity Constraints," *SIAM Journal on Optimization*, 17 (1), 52–77. [2258]
- LUO, Z.-Q., J.-S. PANG, AND D. RALPH (1996): *Mathematical Programs With Equilibrium Constraints*. Cambridge, U.K.: Cambridge University Press. [2258]
- MCKINNON, K. I. M. (1998): "Convergence of the Nelder–Mead Simplex Method to a Nonstationary Point," *SIAM Journal on Optimization*, 9 (1), 148–158. [2246]
- MELNIKOV, O. (2001): "Demand for Differentiated Durable Products: The Case of the U.S. Computer Printer Market," Working Paper, Yale University. [2234,2251]
- NAIR, H. (2007): "Intertemporal Price Discrimination With Forward-Looking Consumers: Application to the US Market for Console Video-Games," *Quantitative Marketing and Economics*, 5 (3), 239–292. [2234,2251]
- NEVO, A. (2000): "A Practitioner's Guide to Estimation of Random Coefficients Logit Models of Demand," *Journal of Economics and Management Strategy*, 9 (4), 513–548. [2232,2237,2245, 2246]
- (2001): "Measuring Market Power in the Ready-to-Eat Cereal Industry," *Econometrica*, 69 (2), 307–342. [2246]
- NOCEDAL, J., AND S. J. WRIGHT (2006): *Numerical Optimization*. New York: Springer. [2239]
- PETRIN, A., AND K. TRAIN (2010): "Control Function Corrections for Unobserved Factors in Differentiated Product Models," *Journal of Marketing Research*, 47 (1), 3–13. [2233]
- ROBINSON, S. M. (1974): "Perturbed Kuhn–Tucker Points and Rates of Convergence for a Class of Nonlinear-Programming Algorithms," *Mathematical Programming*, 7 (1), 1–16. [2242]
- RUST, J. (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica*, 55 (5), 999–1033. [2232]
- (1988): "Maximum Likelihood Estimation of Discrete Control Processes," *SIAM Journal on Control and Optimization*, 26 (5), 1006–1024. [2262]
- SCHIRALDI, P. (2011): "Automobile Replacement: A Dynamic Structural Approach," *RAND Journal of Economics*, 42 (2), 266–291. [2234,2251]
- SU, C.-L., AND K. L. JUDD (2012): "Constrained Optimization Approaches to Estimation of Structural Models," *Econometrica*, 80 (5), 2213–2230. [2232,2241]
- WALTZ, R. A., AND T. D. PLANTENGA (2009): "KNITRO 6.0 User's Manual," Ziena Optimization, Inc. [2247]
- WRIGHT, M. H. (1996): "Direct Search Method: Once Scorned, Now Respectable," in *Numerical Analysis 1995: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*, ed. by D. F. Griffiths and G. A. Watson. Harlow, U.K.: Addison Wesley Longman, 191–208. [2246]

University of Chicago Booth School of Business, Chicago, IL 60637, U.S.A. and NBER; [jdube@chicagobooth.edu](mailto:jdube@chicagobooth.edu),

Dept. of Economics, University of Michigan, Ann Arbor, MI 48109, U.S.A. and NBER; [jtf@umich.edu](mailto:jtf@umich.edu),

and

University of Chicago Booth School of Business, Chicago, IL 60637, U.S.A.; [Che-Lin.Su@chicagobooth.edu](mailto:Che-Lin.Su@chicagobooth.edu).

Manuscript received May, 2009; final revision received October, 2011.